Reflection Report

Submitted by: Yerbol Izbassarov

Full Name:	Yerbol Izbassarov
External ID:	0825CbAT98
Gender:	Male
Age:	38
Submitted At:	2025-04-18 23:26

1. CBC, CBE, and CBA as a System

Competency-Based Curriculum (CBC) defines learning content and outcomes through specific, practical competencies. It goes beyond theory, focusing on real-life tasks and interdisciplinary learning to develop applicable skills. Competency-Based Education (CBE) is an instructional approach where learners move at their own pace, progressing only after mastering the current topic. Instructors act as facilitators and coaches, not traditional lecturers. Competency-Based Assessment (CBA) evaluates learners based on how well they apply knowledge in real-world scenarios, rather than their ability to memorize and learn. It emphasizes practical, project-based, and role-play tasks over standard written tests. These three elements are interconnected: CBC determines what is taught, CBE guides how it's taught, and CBA measures the outcomes. Together, they form a complete and logical learning cycle. In my practice, I effectively applied CBC, CBE, and CBA during a professional development course on "The For loop in the Python programming language." The goal was for learners to understand and use For loops to solve real problems. Instead of just explaining syntax, I gave a practical task: to build a Python program that automatically generates a work schedule. This aligned with CBC by focusing on applied knowledge. Following CBE, learners worked at their own pace. Some advanced quickly and suggested improvements, while others needed extra support, which I provided individually.However, a few struggled with the concept due to limited personal guidance, showing a gap in CBE implementation. I addressed this in future lessons by adding more individual time and examples. This experience showed how CBC, CBE, and

CBA together support deeper learning and skill development.

2. Curriculum Development and Learning Goals

In the context of a Competency-Based Curriculum (CBC), high-quality learning goals, activities, and assessments are closely interconnected. Learning objectives must be specific, measurable, and relevant to real-life contexts. Activities should directly support the achievement of those goals, and assessments must clearly demonstrate whether the objectives have been met. While teaching the topic “The For loop in Python,” I formulated the following SMART-aligned learning objective: “Within 30 minutes, the learner will understand the structure and idea of the for loop and write a Python program that calculates and displays the squares of numbers from 1 to 100. The program must function correctly and output results to the console.” This objective is concrete, aligned with learner performance, and allows for measurable assessment and evaluation. As part of this task, the learner was expected to understand the syntax and logic of the for loop, write the program to calculate some squares, set appropriate parameters (start, end, step), execute the program, and verify the console output. The assessment was based on the following clear and fair criteria: • If any errors were found, the learner was expected to identify and correct the logic, then revise the work; • Whether the learner correctly applied the syntax of the for loop; • Whether the condition was accurately defined (i.e., from 1 to 100); • Whether the output was precise and properly formatted; • Whether the logical reasoning used in the code was sound; • Whether the learner was able to clearly articulate and explain their solution. The learning activities were practical and aimed at building functional literacy. Learners engaged in tasks that required both coding and reasoning. Strategies included pair work, algorithm analysis, and hands-on programming. Assessment followed the CBA model and included peer review and self-reflection. The lesson’s strength lay in real - world application and active learner engagement. However, some learners struggled with logical structuring. In future lessons, I plan to include visual flowcharts and algorithm diagrams to improve understanding of loop structures.

3. Assessment Quality: Validity, Reliability, and Fairness

While developing a test on the topic of branching algorithms in Python for adult learners, several critical aspects of assessment quality had to be considered. The validity of the task-its alignment with learning objectives - was clearly demonstrated. The assignment required learners to input two numbers and determine which one was greater using conditional statements (if, elif, else). This task not only assessed syntactic knowledge but also encouraged logical reasoning, which significantly enhanced its content validity. It provided a comprehensive

approach to evaluate students' understanding of algorithmic thinking. In terms of reliability, all submissions were evaluated using a shared rubric. The rubric clearly defined criteria such as correct use of conditionals, input/output commands, and the accuracy of the output. This ensured consistent scoring regardless of who assessed the work. There were no discrepancies between teachers' scores, as the criteria were both measurable and easy to interpret, contributing to scoring reliability. Fairness and consideration of learner diversity were addressed by designing the task at three levels of difficulty. The basic level involved comparing two numbers; the intermediate level included comparisons between negative and positive numbers; and the advanced level required handling equal values. This tiered structure allowed learners to work according to their individual skill levels and promoted inclusivity. Each level had its own challenges and success criteria, ensuring all learners had a fair opportunity to demonstrate their abilities, regardless of their previous knowledge. In terms of assessment quality, validity, reliability, and fairness were well-integrated. The test effectively aligned with competency-based assessment principles and provided an accurate measure of learners' ability to apply more branching logic in Python. In the future, incorporating visual flowchart-based variations of the task may further improve learners' conceptual understanding and enhance their algorithmic thinking, providing a more engaging experience and knowledge.

4. Grading and Standard Setting

While teaching the topic "Branching Algorithms" in Python, I realized that implementing competency-based assessment significantly enhances the quality of instruction. In my experience, the assessment process was transparent, fair, and aligned with the learning objectives. Sharing the grading criteria with learners beforehand helped them clearly understand what was expected. The principle of fairness was maintained by differentiating tasks based on learners' individual levels. For example, while one learner worked with complex logical conditions, another focused solely on using basic if/elsestructures. The grading rubric included four main components: syntactic accuracy, logical coherence, output precision, and the ability to explain one's reasoning. These criteria enabled a holistic evaluation of learner performance. In setting cut - off scores, task complexity was considered: 60% was the threshold for basic-level tasks, 75% for intermediate, and 90% and above for advanced levels. For instance, if a learner correctly applied the if/else construct but had logical errors, they received around 70%. Conversely, if they handled complex conditions well but had syntax issues, they scored around 80%. This approach allowed grading to be meaningfully aligned with the actual learning outcomes. To further improve grading, I would integrate automated assessment tools such as CodeRunner or Replit. Using platforms like CodeRunner would allow me to quickly identify syntax and logic errors in learners' code. Additionally, evaluating the learner's coding process - especially how they debug and resolve errors - could serve as

an indicator of deeper competency. For example, if a learner identifies and independently corrects several errors during coding, it demonstrates strong problem-solving skills. Finally, I aim to enhance reflection skills by incorporating more peer and self-assessment. Allowing learners to evaluate their own code and analyze each other's work promotes critical thinking and deeper understanding. In conclusion, the grading system was generally fair and aligned with learning goals, yet integrating technological tools, process-based evaluation could further enhance its effectiveness and support learners' programming skill development.

5. Use of Rubrics

Rubrics play a critical role in enhancing the quality of assessment and increasing learner engagement in educational settings. In my professional practice, especially when teaching the Python programming language, I have found rubrics to be a powerful tool for ensuring fair, consistent, transparent evaluation. They help clearly communicate what is expected of learners, guide them in self-assessment, and encourage reflective thinking. For instance, while teaching the topic of "Loop Algorithms," I designed tasks such as writing a Python program to compute the sum of even numbers between 1 and 100 using "for" or "while" loops, and another task requiring learners to calculate the factorial of a number input by the user. To assess these tasks effectively, I developed an analytic rubric with four levels of performance. The rubric focused on several core criteria: correct use of loop syntax and structure, logical soundness of the program, presence or absence of syntax errors, accuracy of the output, and whether the user interface was clear and user-friendly. The rubric allowed for a fair and structured evaluation. For example, a student who implemented the logic correctly but had minor syntactical errors received a score of 75%. This level of detailed feedback helped learners identify specific areas for improvement and encouraged deeper learning. The rubric also helped maintain grading consistency across learners, regardless of who was assessing. A strong rubric should be closely aligned with learning objectives, include measurable and understandable criteria, and be introduced to learners in advance. Furthermore, using rubrics for self and peer - assessment encourages learner autonomy and critical analysis. In my experience, rubrics not only improved students' motivation and performance but also supported a more student - centered and inclusive learning environment. As a result, they contributed to a richer and more effective learning experience and also improved the learning environment for learners.

Digital Signature (CMS):


```
MIINGAYJKoZIhvcNAQcCoIINCTCCDQUCAQExDjAMBggqgw4DCgEDAwUAMAsGCSqGSIb3DQEHAaCCBDEwggQtMIID
```