Journal of Contemporary Educational Research, 2022, Volume 6, Issue 8
http://ojs.bbwpublisher.com/index.php/JCER
ISSN Online: 2208-8474
ISSN Print: 2208-8466

A Three-Dimensional Case System Model for Programming Course

Jinghua Zhang*

Computer Faculty, North China Electric Power University (Baoding), Baoding 071000, Hebei Province, China

****Corresponding author:*** Jinghua Zhang, zjh_yhq@126.com

Abstract: This research aims to construct a case resource library for programming course, which can be used for either teachers' teaching or students' learning. The cases cannot be simply piled up but rather require a systematic planning. The solution to this is to design a case system model. The outcome-based education (OBE) concept is adopted to guide the research, and a three-dimensional case system model matching the course objectives is designed. Under the guidance of the model, the case resource library construction is more planned. Cases based on the model can provide all-round support for the cultivation of students' ability by gradually promoting knowledge and technology, frequently exercising one's abilities, as well as expanding diverse and innovative problems.

Keywords: Programming course; Outcome-based education; Case system model; Case resource library

Online publication: August 12, 2022

1. Introduction

In university programming course, case resources are of great importance. Classroom cases, course tasks, homework, experiments, individual projects, and group projects can be considered as cases because they are all different application forms of cases in teaching. Many pedagogical studies take cases as the basis, such as problem-based learning [1-3], project-based learning [4-7], case-based teaching [8], and collaborative learning [9,10]. The flipped classroom, which has been widely used, also adopts case-related teaching methods in classroom teaching [3,10]. It can be appreciated that cases play an important role in programming course. Among the related studies, the majority focuses on the teaching process and teaching methods, wherein case design is not the main research point [2,3,9,10]. On the other hand, there are also studies that discuss the case design. Some studies discuss concrete cases, such as one or several comprehensive cases [8,11,12], project cases with decomposed sub tasks [1,6,7], or a group of experiment cases [5]. Other studies focus on the ideas and methods of case design to support course content, students' interests or major, and so on [13-16]. However, there is no research on the overall cases of the course.

This paper focuses on the overall cases and the construction of a case resource library for programming course. The goal is to promote students' abilities through these cases. Since programming is a course that requires practice, the cultivation of students' ability is a gradual process that mandates a substantial amount of case support. Although well-designed project cases are significant for the improvement of students' ability, other cases at different levels are still indispensable [17]. Thus, a comprehensive and systematic case system model is required for ability cultivation.

With the popularization of engineering education certification, there have been many course reforms and researches based on OBE [18,19]. The OBE concept is also adopted as it is consistent with the goal of ability cultivation: first, the final outcome of the course is determined, the outcome is then refined to the course ability objectives [20], a case system model is designed, and the objectives are supported through cases.

2. Course objectives based on OBE

Based on the concept of OBE, the final outcome of the course is determined. The programming course focuses on cultivating students' programming ability. The final outcome is that students would be able to master the programming language and programming method, solve practical problems through programming, as well as establish computational thinking. Such an outcome is general; thus, it cannot guide the teaching design. The outcome must be further clarified, through which three questions should be answered.

(1) From the perspective of problem solving, what abilities do students need?
(2) From the perspective of knowledge and technology, what tasks can students complete?
(3) From the perspective of problem characteristic, what kind of problems can students solve which reflects their innovative thinking?

Based on the three questions, the final outcome can be clarified from three perspectives, taking them as the course objectives, namely ability objectives, task objectives, and innovation objectives. This research takes the object-oriented programming (Java) course in the second semester of the first year as the research subject.

2.1. Ability objectives

The ability for solving practical problems by programming is a comprehensive ability. Referring to the Engineering Education Professional Certification graduation requirements, this ability is divided into 6 sub-ability objectives (AOs), as shown in **Table 1**.

Table 1. Ability objectives (AOs) of object-oriented programming (Java) course

No.	Ability objective	Description
AO1	Basic grammar and programming skills	Able to master the programming language syntax and programming skills, program according to the design, and obtain the correct operation results.
AO2	Problem analysis	Able to master simple data structures and basic algorithm ideas, analyze problems, use programming languages to represent problems (data representation and data structures), and use program algorithms to solve problems (algorithms).
AO3	Unit design	Able to understand the correspondence between object-oriented concepts and problem domains, model the things in the problem domain, design object-oriented program units, as well as design the relationship and collaboration between the units.
AO4	Overall design	Able to grasp the ideas and principles of system construction, design the overall program structure to meet functional requirements of the problem, and ensure the reasonability of the program structure.
AO5	Program analysis	Able to learn programming skills by reading programs, understand the design ideas of programs, as well as analyze and evaluate the design schemes of programs.
AO6	Practical problem solving	Able to establish computational thinking and solve practical problems by programming.

Out of the 6 sub-ability objectives, the first five are oriented to different ability aspects, while the last is a synthesis of the first five abilities, which is also the final outcome of the course. Such a decomposition can guide teachers in conducting specific exercises and explanations for different problem-solving ability aspects in the teaching design, so as to achieve the goal of gradually constructing students' abilities.

2.2. Task objectives

Although the final outcome entails the use of programming to solve problems, the types of problems that can be solved are related to the knowledge and methods that students have already learned. Therefore, teachers need to set appropriate task objectives (TOs) supportable by students' knowledge and technology. In this section, the task objectives are divided according to the learning process, and sub–task objectives are set for each learning unit (knowledge and method). With the learning process, the difficulty of sub–task objectives gradually increases and matches the programming technology students should have. For example, the learning content "class and object" can be divided into four learning units and the task objective for each unit is established in advance, as described in **Table 2** (assume that the sequence number of the task objective in **Table 2** starts with i).

Table 2. Task objectives (TOs) of object-oriented programming (Java) course

No.	Learning unit	Task objective
TOi	Class and object A: basic	Solve the problems that a single class can accomplish able to model one thing and support the simple function of the thing.
TOi + 1	Class and object B: object array	Solve the problems that requires a group of similar objects.
TOi + 2	Class and object C: relationship and collaboration between two classes	Solve the problems that two classes can accomplish: able to model two things and support the functions by two things collaboration.
TOi + 3	Class and object D: relationship and collaboration of multiple classes	Solve complex problems: able to model the complex real world and problem domains.

2.3. Innovation objectives

Innovation objectives include using a variety of problems to cultivate students' creativity. The diversification of problems is not attributed to the number of problems. The innovation objectives (IOs) are to enable students to identify characteristic problems, so as to expand their cognition and inspire them to think innovatively. The innovation objectives are divided into sub-objectives according to the characteristics of the problems, as shown in **Table 3**, and each of them represents an application direction.

Table 3. Innovation objectives (IOs) of object-oriented programming (Java) course

No.	Innovation objective	Description
IO0	Cognition of basic concepts and methods	Things that are familiar to everyone can be used for conceptual understanding.
IO1	Information management	Common applications, involving a large amount of information.
IO2	Daily life application	Common applications and tools.
IO3	Small game	Arouse students' interest and enthusiasm.
IO4	Mathematical modeling and calculation	Many majors require strong mathematical modeling and problem-solving skills.
IO5	Statistics and analysis	Statistics and analysis are skills required by almost all majors (phenomena, events, and processes; collect data, text, and images to obtain valid conclusions).

(Continued on next page)

(Continued from previous page)

No.	Innovation objective	Description
IO6	Simulation	Simulate something (events, processes, organizational structures, etc.) that cannot be done or seen, showing its phenomena and evolution.
IO7	Major problem	Questions related to students' major would make the course more useful.
IO8	Other application	Novel problems that do not belong to other classifications.

3. Three-dimensional case system model

The programming course requires a large number of cases to assist students in learning. All cases should be regarded as an organic body. Each case is a part of the body and performs its duty. As a whole, the cases promote students' abilities step-by-step and serve the final outcome.

Based on the aforementioned three-dimensional curriculum objectives, a three-dimensional case system model is constructed, as showed in **Figure 1**, in which the three dimensions form a three-dimensional grid. The red grids represent the final outcome, and each case belongs to a specific grid. Through the model, students would be able to appreciate what level a case is, what abilities the case exercises, and what problems the case would solve. Students would also be able to know where the final outcome is and how to get there.

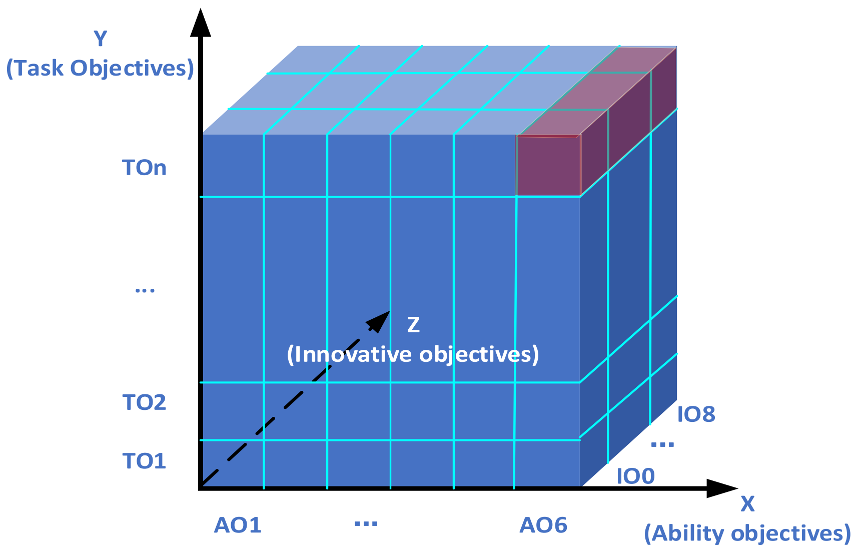

Figure 1. Case system model

4. Case library design

The case library takes **Figure 1** as its blueprint. The y-axis not only represents the task objectives, but also the teaching process. The design scheme of the case library is as follows: the y-axis is taken as the main line, the case group is designed by layer, and the difficulty is increased layer-by-layer.

4.1. Case group design within a layer

Based on the task objectives (TOs), a set of cases is designed for the layer, and the design takes into account the ability objectives (AOs) and innovation objectives (IOs).

(1) Considering the AOs, how does the case support the ability objectives (AO1–AO6)? First, different problems are selected for the case. Some problems are only partial fragments of real problems, while others are integral problem or application. Then, different settings are used to address the problem for different AOs. Strategies for AO1-AO6 are shown in **Table 4**.

Table 4. Case design strategies for the AOs

No.	**Case problem description**	**Problem (integral or partial)**
AO1	Provide design details and concentrate on practicing programming skills.	Partial
AO2	Emphasize on algorithm and data structure.	Partial
AO3	Provide a rough design, focusing on the design of specific program units, relationships, and collaborations.	Partial
AO4	Provide hints for the overall design.	Integral and simple application
AO5	Provide the problem and source program.	Integral or partial
AO6	Students design by themselves using just questions as guidance.	Comprehensive application, Practical problems

(2) Considering IOs, the cases provided in many textbooks are relatively similar, and most of them belong to IO0, IO1, and IO2, which does not achieve the purpose of inspiring students to innovate. There are also some textbooks that provide novel cases, but they are not classified, which is not convenient for students to summarize method for further innovation. This study designs sub–innovation objectives and guides teachers to design richer cases with different characteristics through these directions. **Table 5** shows some examples for each IO.

4.2. Case group design across layers

Along the y-axis, there is an increase in the programming methods and techniques that students master and an expansion in the problems that can be solved; in that case, some consecutive scenarios can be added between layers.

(1) Different implementation schemes are used for the same problem in different layers. Such a set of cases reflects the changes in the problem-solving methods brought about by the changes of the learned knowledge and cultivates students to propose a more appropriate solution to a problem.
(2) A complex problem is divided into several minor problems introduced as the layer rises. Such a set of cases enables students to better appreciate the changes in their complex problem-solving ability.
(3) Combining the two, the problem expands from simple to complex, and the solution also changes with the learning process.

Table 5. Case design example for the IOs

No.	**Case examples**
IO1	Personal address book, schedule management, and user management.
IO2	Calculator, alarm clock, and bank queuing.
IO3	Gobang, guessing numbers, and greedy snake.
IO4	Pure math problems: Fibonacci sequence and equation solving. Real-world math problems: charging problems and planning problems.
IO5	Analyzing the probability of winning the lottery and the relevance of the characters in a book, as well as counting students' grades.
IO6	Simulating coin tossing (100 times), the spread of infectious diseases, and the movement of small balls.
IO7	(Software Engineering major) Simplified version of web application: takeout website and movie website.
IO8	Warehouse robot path planning and sweeping robot path planning,

5. Conclusion

The construction of case resource library is an important part of curriculum construction. In actual course teaching, teachers can adopt different teaching strategies while applying cases. More importantly, the set of case resource itself also allows students to learn by themselves after class. The three-dimensional case system model functions as a guide for students to learn by themselves.

Under the guidance of the system model, the case resource library construction is more planned and targeted. The construction involves continuous accumulation of cases, especially characteristic cases, which may be used for inspiration or reference. In everyday life, work, or reading, if a new case is found, it will be redesigned, tailored, and added to the case library as well as linked to the corresponding model grid, thus gradually enriching the case resources.

Funding

The "OBE-Oriented Programming Course Practice Teaching Research and Resource Construction" Project supported by the Association of Fundamental Computing Education in Chinese Universities in 2021. (Project Number: 2021-AFCEC-246)

Disclosure statement

The author declares no conflict of interest.

References

[1] Ribeiro AL, Bittencourt RA, 2019, Proceedings of the 49th Annual Frontiers in Education (FIE) Conference, October 16–19, 2019: A Case Study of an Integrated Programming Course Based on PBL. IEEE, New Jersey, 1–9.

[2] Ibrahim N, Halim SA, 2013, Proceedings of the 4th International Research Symposium on Problem Based Learning (IRSPBL), July 2–3, 2013: Implementation of Project-Oriented Problem-Based Learning (POPBL) in Introduction to Programming Course. Aalborg Universitetsforlag, Aalborg, 279–288.

[3] Wang G, Zhao H, Guo Y, et al., 2019, Proceedings of the 2019 14th International Conference on Computer Science & Education (ICCSE), August 19–21, 2019: Integration of Flipped Classroom and Problem Based Learning Model and Its Implementation in University Programming Course. IEEE, New Jersey, 606–610.

[4] Saad A, Zainudin S, 2022, A Review of Project-Based Learning (PBL) and Computational Thinking (CT) in Teaching and Learning. Learning and Motivation, 78: 101802.

[5] Doddamani ST, 2018, Project Based Learning of Programming Subject: Case Study on Data Structures. Journal of Engineering Education Transformations, 31(3): 250–255.

[6] Huang H, 2016, The Incremental Teaching Project Design for Project-Based Learning and Its Application in Java Programming Course. Online Submission, 4(6): 191–197.

[7] Ge Q, Ding G, 2012, Proceedings of the 2012 7th International Conference on Computer Science & Education (ICCSE), July 14–17, 2012: Exploration of Project-Based Teaching Content Reforms on Programming Practice Course. IEEE, New Jersey, 1408–1411.

[8] Tan J, Guo X, Zheng W, 2014, Case-Based Teaching Using the Laboratory Animal System for Learning C/C++ Programming. Computers & Education, 77: 39–49.

[9] Echeverria L, Cobos R, Machuca L, et al., 2017, Using Collaborative Learning Scenarios to Teach

Programming to Non-CS Majors. Computer Applications in Engineering Education, 25(5): 719–731.

[10] Hayashi Y, Fukamachi KI, Komatsugawa H, 2015, Proceedings of the 2015 International Conference on Learning and Teaching in Computing and Engineering, April 9–12, 2015: Collaborative Learning in Computer Programming Courses That Adopted the Flipped Classroom. IEEE, New Jersey, 209–212.

[11] Cunningham HC, Liu Y, Zhang C, 2006, Using Classic Problems to Teach Java Framework Design. Science of Computer Programming, 59(1–2): 147–169.

[12] Kang DK, 2010, A Case Study of Puzzle Solving Applied to Programming Practice. Journal of Engineering Education Research, 13(2): 3–6.

[13] Chen R, Xue L, 2008, Proceedings of the 9th International Conference for Young Computer Scientists, November 18–21, 2008: Integrated Application of Project Cases in Programming Course. IEEE, New Jersey, 2625–2629.

[14] Liu Q, Wang X, 2009, Proceedings of the 2009 4th International Conference on Computer Science & Education, July 25–28, 2009: The Anchored Instruction for Programming Courses: Cases, Tasks and Knowledge Transfer. IEEE, New Jersey, 1549–1552.

[15] Zhao L, Su X, Wang T, et al., 2015, Proceedings of the 2015 IEEE Frontiers in Education Conference (FIE), October 21–24, 2015: Interest-Driven and Innovation-Oriented Practice for Programming Course. IEEE, New Jersey, 1–6.

[16] Geng L, 2015, Proceedings of the 2015 10th International Conference on Computer Science & Education (ICCSE), July 22–24, 2015: Teaching Exploration and Reform of Program Design Course for Digital Media Art Students. IEEE, New Jersey, 842–845.

[17] Jazayeri M, 2015, Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, May 16–24, 2015: Combining Mastery Learning with Project-Based Learning in a First Programming Course: An Experience Report. IEEE, New Jersey, 315–318. htttps://doi.org/10.1109/ICSE.2015.163

[18] Ze L, Lang Y, Yanzhong Z, 2021, Proceedings of the 2021 2nd International Conference on Education, Knowledge and Information Management (ICEKIM), January 29–31, 2021: Research and Practice on the Evaluation Method of Objectives Achievement for "Mechanical Design" Course Based on Outcomes-based Education. IEEE, New Jersey, 195–198.

[19] Katawazai R, 2021, Implementing Outcome-Based Education and Student-Centered Learning in Afghan Public Universities: The Current Practices and Challenges. Heliyon, 7(5): e07076.

[20] Gururaj C, 2018, Proceedings of the 2018 IEEE 6th International Conference on MOOCs, Innovation and Technology in Education (MITE), November 29–30, 2018: Defining Course Outcomes based on Program Outcomes and Bloom's Taxonomy for the course on Image Processing. IEEE, New Jersey, 120–113.