A Sample Design in Programming with Four-Component Instructional Design (4C/ID) Model

[1] zaferguney@aydin.edu.tr, egem98@gmail.com, Department of Computer and Instructional Technologies,Istanbul Aydin University, Turkey

Zafer Guney [1]

http://dx.doi.org/10.17220/mojet.2019.04.001

ABSTRACT

The aim of this study is to introduce the four-component instructional design (4C/ID) model in teaching loops for programming by using the model, which is defined in programming instruction, with the principles of the model for teaching technical skills and to discuss the differences in instructional planning and to discuss in the process of design of loops. It is important to integrate knowledge, skills and attitudes in the transfer of knowledge learned in schools to our business life and culture, and to transfer behaviors to new tasks. In the software development process, the planning, design, development and implementation of the programming courses with appropriate instructional design models and strategies should be ensured in order to teach and learn the subjects of the different courses in an effective and meaningful way. At this time, four-component instructional design (4C/ID) model are used for learning complex technical skills and presented in programming to apply loops for teaching skills in software development for programmers, designers and educators. At the end of study, the relationships between ID models and their use in programming process are indicated, and suggestions for developers, programmers and designers are discussed for the future research and applications in programming.

Keywords: 4C/ID model, learning technical skills, learning theories, programming and complex skills

Introduction

In recent years, it is seen that programming and coding trainings are in a process from preschool level to university and after. With the philosophy of educational technology and how instructional design and models can be adapted to enrich social life with industry 4.0 approach and what it can do with artificial intelligence approach, it becomes an important technology problem in the education process. Thus, how to integrate the skills related to these issues in educational technology and instructional design applications; How effective and meaningful manner issues students can gain the programming skills remains on the agenda for a solution as teaching strategies (Guney, 2019) also learning levels and programming skills due to the low of venture capital for the very short time the opportunity to get a share of global capital in the world that is on the front and many small software entrepreneurs with their applications to the world with the giant of the world. It is also observed in the results of the study that the program writing and coding education has effect in terms of positive cognitive, affective and social adjustment differences (media use skills) on learners rather than just program coding. It is important to integrate knowledge, skills and attitudes in the transfer of knowledge learned in schools to our business life and culture, and to transfer behaviors to new tasks. In the software development process, the planning, design, development and implementation of the programming courses with appropriate instructional design models and strategies should be ensured in order to teach and learn the subjects of the different courses in an effective and meaningful way.

Problem:

The development of the programming courses described in this study and the acquisition of the information defined as technical skills are an important process for the learners to acquire programming skills. For this reason, in order to gain cognitive (technical) skills in programming teaching, the four-component instructional design (4C/ID) model can be used in programming as a technical skills will be discussed (van Merriénboer (1997).

In this context, it is necessary to point to the example steps followed for programming training. The instructional design (ID) model can be used to learn a selected subject related to the course /software development and teaching in accordance with the learning strategy. This process is considered as an instructional design (ID) approach, which is compatible with the multimedia project design, production and evaluation model in education. In this context, the subject of loops was taken as a sample subject in the process of learning software development.

In programming teaching, it is necessary to make planning within the framework of concept teaching and to know why basic symbols and concepts are used (Guney, 2019). For example, when the loops concept is taught and the work is done to see how it works in many programming languages, the loops process causes a loss of time and effort. However, it should be emphasized that the concepts should be taught more than the syntax of any language. After learning the concepts, coding in the desired language will be more practical (such as C, C ++, C #, Php etc.). Thus, with the principles of instructional design (ID) approach of this process, it is a learning problem that needs to be a series of requested operations in order to develop an effective learning environment, overlapping with the solution of waiting and technical skills. (Guney, 2019; Ipek, Izciler, Baturay 2008)

Purpose of the Study:

The aim of this study is to introduce the four-component instructional design (4C/ID) model in programming loops by using the model, which is defined in programming instruction, with the principles of the model for teaching technical skills and to discuss the differences in instructional planning and to discuss in the process of design of loops. In addition, the ten steps proposed for complex learning (van Merriënboer and Kirschner 2007) and the ten steps of a different approach to instructional design, complex cognitive, high-level, algorithm based, limited coding, the steps, techniques in the acquisition of technical skills and four steps in a basic framework presented. The frame for planning and the design of the process is to offer a new approach to program developers, trainers and designers (van Merriénboer, 1997; van Merriënboer and Kirschner 2007).

Importance of Study:

In addition to the difficulties experienced in teaching in the studies related to programming teaching, the most discussed factors are gender, personality, intelligence, attitude towards computer, experience, comfort level, mathematics background, lessons learned, game play and other factors (Charlton and Birkett, 1999; Wilson, 2002). However, there is not much mention of learning strategies and the application of instructional design models for teaching effective and meaningful programming. Therefore, in this study, it is revealed that the contribution of instructional design approach and models for the transfer of knowledge can contribute to programming education effectively. Designing a new learning environment for the subject of loops in the programming process with the Four-Component Instructional Design Model (4C/ID), this teaching process is effective, meaningful and efficient in acquiring different technical skills, and it is also important for programmers, designers and educators.

Limits of the Study

Four-Component Instructional Design (4C/ID) model of the comprehensive four stages and the horizontal and vertical intersection of these steps include analysis of skills, limited coding, methods and events in the model, ten steps in programming, the teaching of the concept of loops in a basic framework, the design stages it is thought to be used step by step.

Many difficulties in the process of programming teaching arise in terms of learning and teaching of technical skills. This study, as a technical skill, involves only introducing examples of loops by introducing examples of loops in programming, illustrating how a teaching design approach can be used by experts to develop programming and technical skills for future developers and designers.

Loops: It is a system that is used to run multiple program segments in a row. Being able to understand the concept of loops in programming language requires different learning and thinking abilities in students. These are defined in the following paragraphs.

Creating Cognitive Processes, Automation and Schemas

In this study and other studies, the effective direction and steps of the 4C/ID model are discussed among the learning processes and the programming processes and learning processes (Guney, 2019; İpek, 2004). Here, examples of how the theme of loops can be designed and modeled in the programming by establishing relationships between the programming process and learning. For example, algorithmic methods, preliminary information, part of the application and presentations are used to create learning environment rules for creating automation and schemas in the programming. For the acquisition of schemas, supportive information, detailing and comprehension, artistic methods and all of the tasks are performed in the separation of skills. It is important that verbal information and operational knowledge are based on psychological and cognitive models in terms of the place of human knowledge.

The information based on the declaration refers to the objects that are presented, the objects, the relations between them and what we know in short. Operational information processes, on the other hand, indicate how little and very effective the information is, not the correctness or misunderstanding based on how we know and think. For this reason, verifying procedural knowledge is more difficult than declarative knowledge. When the use of loops in programming is considered as an example, the information and type of the concept of the loops can be explained by word, but it is also used effectively or less effectively in the programming process according to the rule. In this process, concept map, propositions are shown in schema format. Operational information expresses cognitive processes and activates our representations outside the world. It refers to conditions and actions, among which the programming symbols are treated, for example, IF and THEN. As a result, many examples of these can be taught in the form of information collection within the specified ID model by describing the automation and schemas in the loops types. These processes allow the application of the model to analyze and analyze complex cognitive skills during the design phase. The cognitive analyzes also initiate the provision of rule automation and schemas with verbal notification and terminate with the product. For example, analysis of loops types can express events, actions and functions in the form of an impact, concept, plan and principles, target and plan hierarchy. Thus, transactional information is acquired as a product based on target and plan.

What is Instructional Design?

The concept of instructional design (ID) covers a process as a field and discipline. Therefore, as well as student-centered, the problem of the learning problem as well as the systematic and systemic approach to the solution of the problem identified is of special meaning. In this context, different definitions were made by different field experts. According to these; The ID is a systematic and responsive process for the transfer of learning and teaching principles for teaching materials, activities, information sources and assessment plans (Smith & Ragan, 2005). In another definition; ID is based on what we know about management information systems, system design, teaching and learning theories (Morrison, Ross & Kemp, (2001). According to the designers of another approach that is effective for both classroom teaching and project management. Instructional design is defined as “the process of solving learning problems by systematic analysis of learning conditions” as discipline and process (Seels and Glasgow, 1998). Similar definitions of ID were discussed in another study (Guney, 2019).

Preparing teaching instruction in accordance with instructional design models will enable high-level thinking skills that will contribute to the effective, efficient and engaging of teaching. High-level thinking and learning skills; It is a process that determines the lifestyle (culture) and social interaction which constitute the process of problem solving and decision-making according to the purposes, which involves many mental processes based on inquiry. For this reason, the design skills in the design model and the design of the tasks in the teaching programming, the sequencing of the works to be done, the gaining of supporting knowledge, the analysis of the cognitive and mental paths as well as the methodological knowledge, the ways of solution of the cognitive rules, design with the teacher applications in the process of teaching the loops are defined. Thus, students can realize complex skills in adaptation and transformation behaviors at the stage of learning, for example, schemas and loops as concrete technical skills.

Cognitive Loading Theory

The ID theories are the guiding principles of the teaching method (Reigeluth, 1999). The aim of cognitive load theory is to develop instructional design steps based on human cognitive architecture model (van Merriénboer and Sweller, 2005). This architect approach accepts the existence of limited short-term memory (STM), the presence of cognitive schemas in non-limited long-term (LT) memory. Within this framework, learning is defined as a structure and automation within the scheme. Therefore, the programming process has to take advantage of the design principles and strategies in cognitive processing theory. The diagrams defined in the 4C/ID model allow for the elaboration of complex thinking in the design of the programming process, and the ability to learn and implement the technical skills required for this and even offer the opportunity for automation. The operation of this programming is indicated below (Guney, 2019).

Four-Component Instructional Design (4C/ID) Model and Programming Process

Programming skills also require technical and analytical skills, such as thinking, analyzing, replacing values, calculating necessary variables, and establishing relationships between them. For example, in programming, the subject of loops is based on the repetition of the necessary tasks, and the repetition of the skills. For this reason, new loops are written and coded from simple to difficult variables. This also involves loops tasks and tasks, such as an automatic operation, and reinforces repetition. The supportive nature of the information for the software process is effective in creating cognitive skills and mental model. Supporting knowledge creates a bridge in the design of learning tasks with the knowledge of the learners and provides the student with a clear understanding of the loops of the programming that they will be prepared for. As a methodological approach, the student repeats his / her technical knowledge and skills, proceeds step by step, and continues the specialization of the loops. Methodological knowledge can provide the learning of the routine aspects of the learning tasks carried out by the learners, and the place and time should be presented when the student is in full need. Learners are expected to gain behaviors as they become experts. This way, it can make the analytical concepts more functional and easily demonstrate its technical skills in the form of loops. With partial task practice, it is necessary for ordinary learners to have a lot of practice in order to improve the habits of learners. The relationships between the skills to be taught and the relationships between the processes to be performed are shown as components within the framework of the model (Figure 1 and Figure 2) (Guney, 2019).

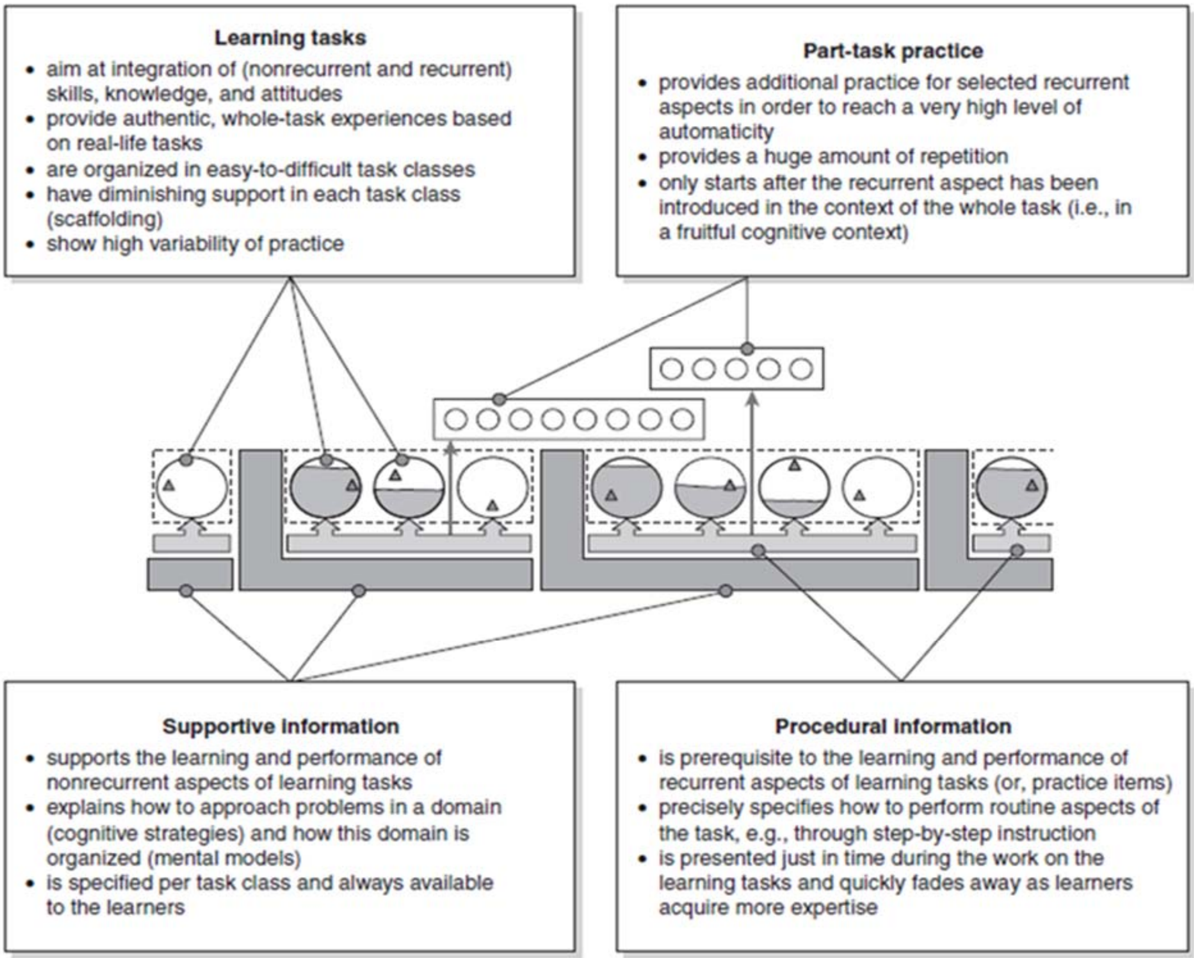

Figure 1. **(4C / ID) Model and A schematic teaching design taken from van Merriënboer and Kirschner, 2007).**

Basic Components of 4C/ID Model and Ten Steps for Complex Learning

Basic Components of the Model	**Ten Steps to Complex Learning**
Learning Tasks	1. Learning Tasks for Design 2. Sequential Task Classes 3. Setting Performance Targets
Supporting Information	4. Supporting Information Design 5. Analyzing Cognitive Strategies 6. Analyzing Mental Models
Methodology	7. Methodical Information Design 8. Analysis of Cognitive Rules 9. Prerequisite to Analyze Information
Partial Task Applications	10. Partial Task Application Design

Figure 2. Steps in teaching programming skills with Four-Component Instructional Design 4C/ID Model (taken from van Merrienboer and Kirschner, 2007).

Learning by Four-Component Instructional Design (4C/ID) Model

Instructional Design Approach and Complex Steps For Complex Cognitive Skills

Here, the design and creation of learning environments for complex cognitive skills is discussed. In the teaching system for teaching any skill, differences occur between the design of the presentation of knowledge and the design (practice) of the application. The loops also include an application between the information given in the programming (software) process and the variables.

The second step concerns the problem-solving process of systematic approaches. This performance is all complex skills or their meaningful perspectives. The heart of this model is the application design of all tasks. The samples studied in the design period, the problems in different formats are informed to the student for each case study in micro level. In this section, briefly, the following issues come to the fore.

The Concept of Loops in Teaching And Instructional Design

It is important to explain the concept and content of the loops for programming at each level. Systematically, events should be sorted, explained, and logically processed. These elements are important steps of the instructional design model. In the following example, the loop consists of seven steps which are given in figure 3. After learning the necessary skills for each step, the proposed scenario will be reached. This step has the meaning of planning in the ID approach. Because ID starts with a plan and continues with the requirement analysis of the model (Seels & Glasgow, 1998).

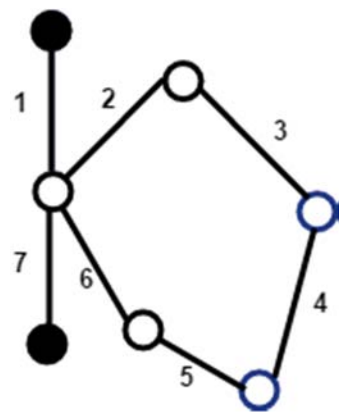

Figure 3. Task steps for the loop

According to the model, the course content is divided into four or five task classes. Task classes are moving from simple to complex. Students start at the simplest level and, as their level increases, they begin to move towards difficult task classes. Each task class content is created with four basic steps and six complementary steps. The steps of the model are shown in schematic form after planning the weekly course contents. In this context, the acquisition of the knowledge and skills required for the programming is adapted in the form of acquisition A and recovery B and is given in table 1. Learning strategies for the design and functioning of the loop are shown with examples and discussed with examples of loop in the coding process.

Table 1. Gain, success criteria and the concept of loop in the programming process

EARNINGS	SUCCESS OF SUCCESS-behaviors and operations		Related section in model
A	INFORMATION	Understands the logic of the loop statements.	Understand the logic of algorithmic detail - perform tasks and analysis
		Understands the measures to be taken in order not to enter the infinite loop	Uses algorithmic methods and uses the methods required for loops.
		Sorts loop statements	To do information analysis, to learn by using the basic statements. Can start coding.
		Sorts the rules for selecting the loop statement to be used by the application.	Learns the requirements for coding and sorting requirements for coding and software.
B	SKILL	It uses the FOR loop.	Writing the corresponding loop. for coding
		It uses the WHILE loop.	Provides the ability to write loops for coding.
		It uses the DO WHILE loop.	Apply the ability of continuity with the concept of continuity in the loop.
		It uses the FOREACH loop.	Writes and produces a new loop by performing more complex problem writing skills.

1. Step: Designing learning tasks

In this step, a holistic form of the tasks that students will see during the training is gradually formed (Figure 4). A set of learning tasks provides the backbone of an educational program for complex learning. The model proposed here is presented for the separation of information required for the design of tasks, as well as the concept information and the information necessary for programming in the loop concept. In the programming course, the student is informed about the purpose and logic of the loop. It fulfills the learning tasks given below. Evaluation of the transfer of information by giving an event at the end of each work done. The purpose of instructional designers is to identify skill requirements and to create appropriate tasks for these skills. Also the aim of the student is to solve increasingly complex problems without help (except just in time during the work). End of the ten steps, these goals will be achieved and showed as in figure 5.

Figure 4. Learning process task steps

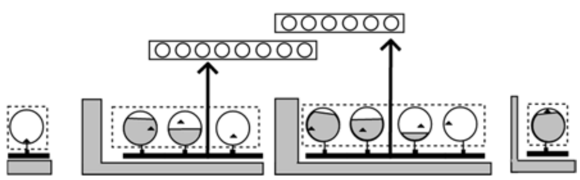

Figure 5. Partial tasks repeated with small circles

Learning tasks:

- Purpose of use of loops and major loops statements
- General working logic of loops

- Infinite loop and endless loop prevention ways
- Learning the For, While, Do While and Foreach loops.

Variability: It is important that the chosen learning tasks are different in all the different dimensions in the real world and they are given in figure 6.

Figure 6. Different dimensions of the topic of designated learning tasks and loops

2.Step: Sequential task classes

From the beginning of an education program, it is not possible to use very challenging learning tasks with high level demands on coordination. For this reason, students start working on relatively easy learning tasks and progress towards more difficult ones (van Merriënboer, Kirschner, & Kester, 2003). The sequence of sequential task classes representing the easy-to-difficult versions of tasks is shown below in the loop topic (Figure-7).

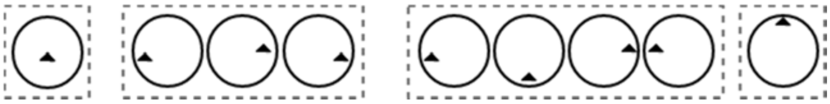

Figure7. Difficulty levels of learning tasks

Task classes, topics and degree of difficulty

- The purpose of the use of loop and the main loop statements
- The general operating logic of loops
- Infinite loop and describe ways to prevent infinite loop
- Difficulty levels are increasingly increased according to tasks. The implementation of the loop process with project applications is illustrated by way of example.

3. Step: Performance goals

In order to decide when students can move from one task class to another, we evaluate the performance and determine the standards that should be done to achieve acceptable performance and provide feedback.

3.1. Performance objectives and indicators

a) Perform the needs analysis of the loops.

b) List the loop keywords and their concepts.

d) Understands the logic of the study of loop statements.

d) Does not need to enter the infinite loop while using the loop statements.

e) Performs the desired algorithm using **For**, **While**, **Do While**, and **Foreach**.

f) Select the appropriate loop statement in accordance with the assigned project task, realize and evaluate the project.

3.2. Project example

Write down a C# program that redirects a user-specified number to the computer (the number will be determined by the user). The software will be forecasted and the user will navigate the keyboard with D (Down) - U (Up). Press O (OK) to confirm that the program will be terminated and the number of times. The learning outcomes of the performance objectives will be demonstrated by the model approach.

4.Step: Supporting information design

When students start working with a new, more difficult task class, it is essential that they receive support and guidance to coordinate different aspects of their performance. Each task class begins with a high level of guidance and support, decreasing towards the end of the task class in figure8. This stage can be expressed as an analysis of tasks and tasks in the traditional instructional design model. Grayscale expresses support rates in teaching. The steps are continuous but the support rate decreases as the expertise is gained. This phase is followed by teaching analysis. Product-oriented: It is solution-process-oriented to deliver products and objectives in education: The goals of achieving the tasks are as seen in figure 8.

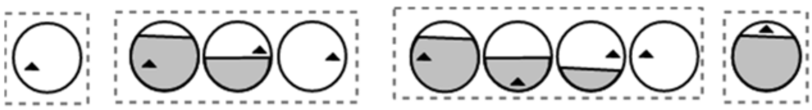

Figure 8. Supporting information design goals.

4.1.Loops

Loops are among the most needed syntaxes in programming languages. If there are business processes that must be performed repeatedly in the program flow, these business processes are performed with the help of loops. There are 4 kinds of loops in C #: For, while, do while and foreach loops. It includes the tasks to be performed here.

4.2.Flow order and operation logic of loops

Loops in the programming language are structures that allow the command lines to be run repeatedly. Let's examine the logic of the algorithm over the "For" loop. The "for" loop is expressed in three parts. In the first of these expressions, an initial value is given for the loop variable. In the second statement, a condition is determined depending on the value of the loop variable. The third expression is often used to increase or decrease the loop variable. Based on the flowchart, Figure 4 shows a flowchart with learning strategies to be followed in loops.

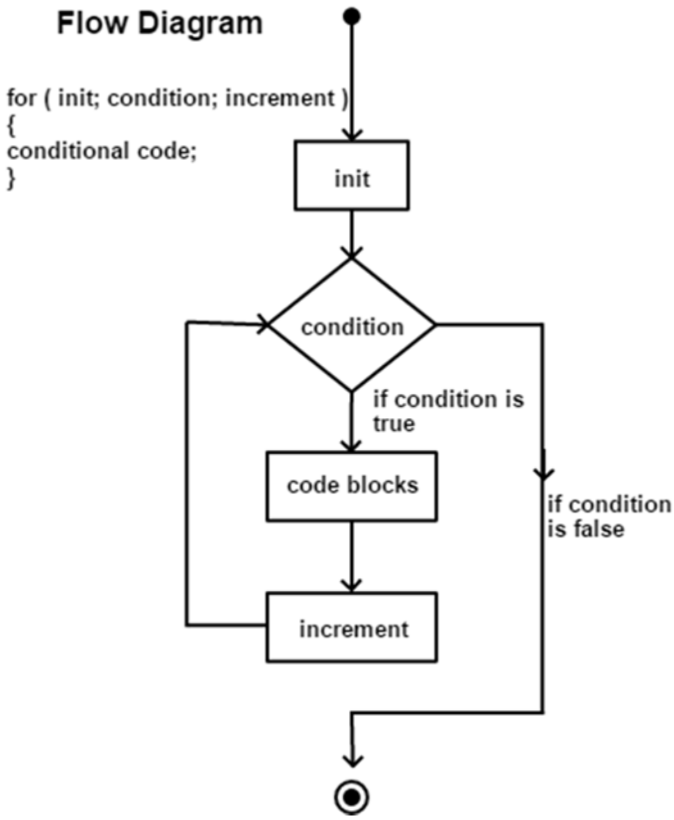

Figure 9. A sample flow chart for loops

4.3.For loop function

In using the "For" = loop, a variable is initially created in the loops, which is the initial value, for example int i = 0 . Then the condition expression that determines the boundaries of the loop comes, for example i <10 . Finally, the mathematical expression takes place above the variable, such as i ++ . In this example, the initial value increment occurs as long as the required condition is true. *The* general format of the *for* loop is as follows: Here is the introduction of the “for” command in the loop and its values. This step is shown in Figure 3 of the model. It is related to the step and reflects a sub-task and features of the algorithmic structure. In the steps followed in the other loop below, the configuration for other processes of the four-component instructional design can be configured in the same manner.

```
For (int i = 0; i <10; i ++)
{
  Console.WriteLine (i);
}
```

The initial value (i) is also referred to as the control variable. The control variable acts as a counter in the loop and is generally used as the value to be checked in the condition. The condition is the mechanism that controls whether the operation to be performed at any time of the loop will continue. The automation function of the model is provided here. The mathematical operation refers to the process to be applied on the initial value (i) each time the loop occurs. If this operation is not performed on the initial value controlled under the condition, an infinite loop occurs. As a result, it accelerates learning by making the inductive function meaningful because it can eliminate the function of maintaining and acquiring the technical skills of the model.

5.Step: Analysis of cognitive strategies

Cognitive strategy is the situation where the student goes to his / her own practice in the situations of remembering, thinking, dreaming, coding, storing, learning, attention, recall. At this stage, the different ways that the student uses to control the thinking and learning process are analyzed. The supporting information required to learn the practical aspects of the learning tasks identified in steps 5 and 6 should be given in such a way that they can combine with their previous information and add new information. This information is indicated at the beginning of the task and must always be available.

6.Step: Analysis of mental models

The mental model is actually a view of the world . In its simplest form, the mental model refers to a number of tools that you use to think. Each mental model presents another frame, so you can look at the world or a certain problem in this framework. At this stage, we have to analyze the different angles that students can look at the problem.

Once the information is matured, our new mental frameworks will begin to emerge after placing the loops in the first mental frame in our memory in order to build on the existing knowledge. Specifically, after building the foundation of the building, we build the same number in each number of bricks to build the wall until we reach the corners, this is a loop. Within our mental models, the frame can be easily arranged. In the same mental frame, this time we can increase the one-to-one aggregation process by incrementing it in the programming algorithm, because the same operation will be performed until it reaches the face. When we construct the same cognitive frame and the wall of the third floor, the mental frame of making similar circular brick strings will be placed in the frame of the first brick sequence that we have created a step before. In our programming process, when we want to list the data, printing of the data in the table by increasing the number of brick numbers in the same mental frame would be easier to produce, and the mental frame of printing with a short code instead of printing individually for each brick would be easier.

7.Step: Methodical knowledge design

Supporting information is usually presented earlier, as it relates to all learning tasks within the same task class. This information is kept in the class in which the students begin to work. Schematically, in figure 10, the gray markings indicate increasingly decreasing support ratios.

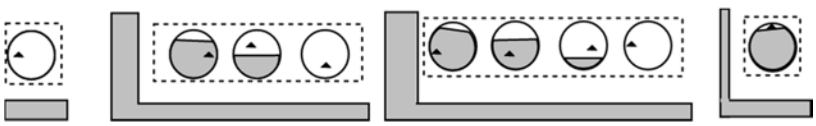

Figure 10. Support information level distribution decreases within learning tasks.

Methodological information is presented at the time (i.e just in time) that students need during their work on learning tasks. As this information disappears when it enters the next learning task, the level of support for new information will be higher as shown in figure 10.

8. Step: Analysis of cognitive rules

In eight and ninth steps more routine learning tasks are analyzed in depth. If there are useful instructional materials such as reference guides or electronic performance support systems, step 7 may be limited to updating these materials and associating them with appropriate learning tasks (van Merriënboer and Kester, 2005). Steps 8 and 9 may then be ignored. However, if methodological information needs to be redesigned, it may be useful to perform step 8; where, it is by analyzing the cognitive rules that specify routine action pairs that direct routine behavior. In step 9, it analyzes the information defined as a prerequisite for the correct use of cognitive rules. The results of the analyzes in steps 8 and 9 are the basis for the design of the methodological information. Finally, depending on the nature of the task and the knowledge and skills needed to accomplish this, it must take the tenth and final step (Methodological information is not considered here since it is newly designed).

9.Step: Prerequisite analysis of information

At this stage, using the similar paths for analyzing the information of the tasks followed in the second step (task classes, subjects and difficulty levels). The logic of the work of the loop statements as learning output of the project is grasped and the rules of selecting the loop statement to be used according to the application are fulfilled.

10. Step: Partial task application design

The sequence of learning tasks may not provide enough repetition to reach the specified level. For the recurring aspect of the task, a partial task application should be performed. The information that needs to be repeated which constitutes the basic skeleton of the knowledge to be learned, is given to the students in the second and third terms. By performing the applications within the steps, it is reinforced by partial tasks in which the learning takes place in all ten steps (Figure 11). Cognitive rules gain power in every successful practice.

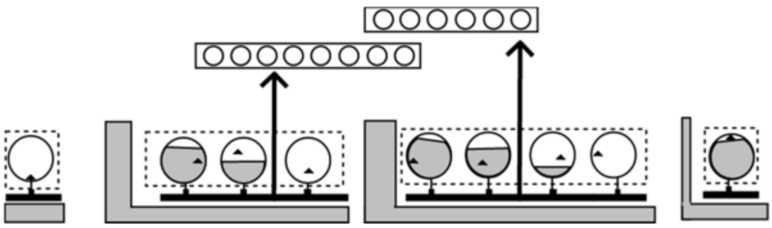

Figure 11. Partial tasks repeated with small loops between the stages of the model

Conclusions and Suggestions

In this study, 4C/ID model techniques and complex skills are developed and examples of the proposed strategies and steps for this model are presented in the programming. This model is based on two basic procedures. One of them is the implementation of the model based on deep expertise and better performance. Second, model-based learning environments are thought to be superior to traditional learning environments. The characteristics of the model, which the numerical and mental thought contributes to the development of technical skills, such as analyzing, providing automation and creating schemas, are also important for concepts and mental skills that are fundamental in the learning and implementation of loops.

As with the analysis of information and tasks in the second step of the model, mental models are crucial for the performance of complex cognitive skills (Guney, 2019). Learning about the subject of loops allows the analysis of similar information and task in terms of the functioning of various loops and the differences between them. Gaining programming skills and analysis of complex skills is feasible by designing and developing software for different environments of learning in the software environment (Guney, 2019; İpek, 2004).

For the complex cognitive skills that do not persist as an instructional model approach of the 4C/ID

Model, it gives designers the opportunity to solve problems and learn more and to be more careful about their transfer (Guney, 2019; İpek, 2004). Some of the loops are continuous and continue the process. Some terminate the process for the given input and value. For the complex learning process, this model explains the ease of use of many suitable models in terms of learning skills, learning problem solving function. The main thing is to design all of the tasks in learning all the ongoing or non-cognitive skills. One of them is the problem solving process in the realization of the loop of the individual. As mental models, what is the problem, how it is organized, and how a work is done are the subjects related to important teaching processes (Guney, 2019). In this respect, the concepts of model and software development process discuss a new approach as a thought by revealing the importance of industrial institutions, businesses and organizations for the solution of problems that require complex cognitive skills as well as scientific techniques, tactics and suggestions for designing effective learning environments.

As a result, with an instructional design model, which can be used for gaining complex and cognitive skills, the operation, structure and techniques that can be used in vocational and technical education are explained with concepts. In this process, the ID model discussed in this process is based on new technologies and the relationships between the software system and behavioral-cognitive and constructivist teaching approaches (moderate-radical) was tried to explain as possible and possible relevance, thought development skills have been revealed. Thus, it has been pointed out how this model can be used to realize permanent, effective and interactive teaching in the field of programming. This study, in short, opens up a new approach to software developers, designers and educators in terms of software development, in the near future, in terms of the functions of instructional designers and technologists and the effectiveness of software development instruction in industry and schools, as well as the acquisition of new complex, mental and cognitive skills.

REFERENCES

Alvarez, C., Brown, C. & Nussbaum, M. (2011). Comparative study of netbooks and tablet pcs for fostering face-to-face collaborative learning. Computers in Human Behavior, 27, 834-844. doi:10.1016/j.chb.2010.11.008.

Branch, R. M.& Dousay, T. A. (2015). *Survey of instructional design models*, Fifth edition, Indiana, AECT.

Charlton, J. P. & Birkett, P. E. (1999). An integrative model of factors related to computing course performance. *Journal of Educational Computing Research, 20(3)*, 237-257.

Guney, Z. (2019). Four-Component Instructional Design (4C/ID) Model Approach for Teaching Programming Skills . *International Journal of Progressive Education, 15*(4), 142-156. doi: 10.29329/ijpe.2019.203.11

Ipek, İ. (2004). Karmaşık Bilişsel Teknik Becerilerin Öğretilmesi Sürecinde Öğretim Programı Tasarımı: Dört Öğeli-Öğretim Tasarım Modeli. IV. International Educational Technologies Symposium. 24-26 November 2004. Sakarya University. Sakarya.

Ipek, I., Izciler, M., & Baturay, M. H. (2008, May). Considerations for integrated e-learning courseware design and instructional design & technology (IDT) approach. In *VIII. International Educational Technology Conference, Proceedings* (pp. 508-512).

Januszewski, A. & Molenda, M. (2008). *Educational Technology, "A definition with Commentary",* NY: Lawrence Erlbaum Associates.

Morrison, G. R., Ross, S. M. & Kemp, J. E. (2001). *Designing Effective Instruction. (2nd Ed.)* Upper Saddle River, NJ: Prentice-Hall, Inc.

Reigeluth, C. M. (1999). *Instructional design theories and models: A new paradigm of instructional design (vol.2)*. Mahvah, NJ: Lawrence Erlbaum Associates.

Reigeluth, C.M., ve Merrill, M.D. (1984). *Extended task analysis procedures (ETAP): User's manual*. Lanham, MD: University Press of America.

Seels, B. & Glasgow, Z. (1998). *Making instructional design decisions (2nd Ed.)*, Upper Saddle River, NJ: Prentice Hall, Inc.

Smith P. L. & Ragan, T. J. (2005). *Instructional Design. (3rd. Ed.)* John Wiley& Sons, inc.

van Merriénboer, J. J. G. (1997). *Training complex cognitive skills: A four-component instructional design model for technical training.* Englewood Cliff, NJ: Educational Technology Publications.

van Merriënboer, J. J. G., Kirschner, P. A., & Kester, L. (2003). Taking the load of a learner's mind: Instructional design for complex learning. *Educational Psychologist, 38(1),* 5–13.

van Merriënboer, J., & Kirschner. P. (2007). *Ten steps to complex learning: A systematic approach to Four-Component Instructional Design.* NJ: Lawrence Erlbaum Associates.

van Merriënboer, J. J. G., & Kester, L. (2005*). The four-component instructional design model: Multimedia principles in environments for complex learning.* In R. E. Mayer (Ed.), The Cambridge handbook of multimedia learning (pp. 71–93). New York: Cambridge University Press.

van Merriënboer, J. J. G. & Sweller, J. (2005). Cognitive load theory and complex learning: Recent developments and future directions. *Educational Psychology Review, 17,* 147–177.

Wilson, B. C. (2002). A Study of Factors Promoting Success in Computer Science Including Gender Differences. *Computer Science Education, 12(1-2),* 141-164.