

Computational and Soft Skills Development through the Project Based Learning

Innokenti Semoushin, Julia Tsyganova, and Vladimir Ugarov

Ulyanovsk State University, 42 L. Tolstoy Str., 432970 Ulyanovsk, Russia
{SemoushinIV, TsyganovaJuV, UgarovVV}@ulsu.ru
<http://staff.ulsu.ru/semoushin/>

Abstract. In this paper we discuss some experiences gained by Department of Mathematical Cybernetics and Informatics, Ulyanovsk State University, Russia, on teaching computer science and mathematics courses for students majoring in Computational Science and Engineering (CSE).

1 Introduction

The demands for experts in Computational Science and Engineering (CSE) call into being different educational approaches such as new didactic environments [1], [2], new academic programs [3], [4], and new learning methods supported by technology [5]. Nevertheless, a building block to educate CSE professionals remains as before: highly developed skills of algorithmic thinking and computer programming. A promising development in university education involves the application of project-based learning as a curricular vehicle to develop students' computational talent. We define the Project-Based Learning (PBL) as a teaching paradigm stating that the student has perfectly understood a computational method not until he or she becomes able to make the computer to do all the work prescribed by the method. We express this by a piece of advice we give in jest to some of our students: "If you argue that you know the method well, please teach the computer to do the same".

2 Employers' Expectations

Training experts in CSE as well as in "Applied Mathematics" (AM), "Information Systems" (IS), "Information Technology" (IT) and some related fields assumes the profound study of computer programming. It is well known that contemporary employers expect not simply programmers but professionals having depth in one specific area of programming as well as breadth in computer science in general and in other fields relevant to information technologies, such as:

- Database Design and Management
- Data Communication (Computer) Networks

- System Programming (system management and control software in a variety of hardware and user environments)
- Application Programming (development of the problem-oriented applied programs)
- System analysis (finding the effective solution to the task and program specification development, on the basis of customer requirements)
- Etc.

Knowledge in these areas is requisite to carry out the main task – maintenance and development of information systems in industry and academic setting. One of the principal disciplines necessary to prepare experts for careers in CSE and IT is Programming. The purpose of this discipline is not only to develop in students the strong theoretical background but to teach the solid practical skills in computer programming. The students majoring in CSE must gain actual experience in solving problems through writing, entering, and running programs in order to develop their ability of structural algorithmic thinking. They also need to consider versatility of the discipline of Programming and its deep relation to many topics of mathematics: algebra, geometry, calculus, etc.

3 Traditional Approaches to Teaching Programming

Over the years many different techniques for teaching Programming and other related courses have been devised. Leading scientists, such as N. Wirth, E. W. Dijkstra, W. M. Turski and others paid much attention to the problem (see for example, [6], [7], [8], etc.).

Up to now, many educational courses in computer science are based on the methodological principles of teaching programming developed by Dr Niklaus Wirth, the famous Swiss scientist. Algorithmic language Pascal created by him, has been world wide practised, especially as the training programming language [6]. E. W. Dijkstra [7] has offered a method of formal program derivation from mathematical formulation of a problem, and some new approaches to decomposition of algorithms. Now Programming is the base of study at mathematical and engineering faculties of universities. The great experience has been accumulated, the various approaches to and methods of teaching this course differing on a goal orientation have been considered and approved. For example, the method suggested by Kaimin [9], is aimed at the developed student logic thinking and general computer literacy, allowing a student to work with ready-made applied software packages. The primary preparation method used at the Mechanics and Mathematics Faculty of Moscow State University (Russia) is aimed at establishing the base of general programming culture of the students [10]. This method is built upon developing the mini-projects to consider and make use of the basic propositions of programming. The issues of computational mathematics, numerical methods and other subjects of mathematics are not considered in this course. The material is based on concept of the “Executor”, which can carry out “Actions” or “Instructions” over “Objects” to change their “States”. The

set of “Actions” refers to as “System instructions” of “Executor”. This method gives the basic attention to consideration of algorithms, not especially becoming attached to the certain programming language.

4 Programming as a Sort of Engineering Design

4.1 Aiming at the Main Professional Qualities

Along with the other above mentioned methods, we consider the course “Programming” as a sort of engineering design. By definition of John R. Dixon [11]: the engineering design in essence is not the “art”, but it represents activity which one can investigate and analyze, and master its bases during training... [11]. An expert in programming should correctly solve a lot of problem characteristic for design process while developing a software package. As a rule, the aim consists in the development of the software product able to carry out the job subject to a number of restrictions imposed on the product features. It is known that design is essentially iterative process consisting of successive stages on the way to the desired product some of which are to be repeated in order to improve or update the project or to correct the errors. The earlier corrections are made, the less amount of work and expenses will be spent for the whole project. On each design stage there exists a problem of a choice between many variants of the further work. This assumes decision making to satisfy one or other given criterion. In the designing process, it is very important to understand correctly the diverse requirements imposed on the future product. Consequently, along with the good knowledge of programming process, it is necessary to know well the subject area for which the project is developed. From the above brief reasoning about the design process it is clear that the modern programmer must be possessed of many qualities:

- High qualification in programming.
- Ability in doing analysis of a product on various design stages for revealing the elements having influence on the quality.
- The sufficient knowledge of the subject area for which the project is developed.
- Strong Computational Mathematics knowledge in designing solutions to mathematical and computer oriented problems.
- Thinking skills in making decisions under conditions of uncertainty in view of all essential factors [5].

The development of such qualities in prospective computer professionals is the basic task of the university courses “Computer Science” and “Programming”.

4.2 The Basic Phase of Learning

As these courses are read at the first year of study and are basic, from the classical point of view it is necessary to keep elements of directive training.

Because programming is the practical discipline having, on the one hand, the strict language constraints and on the other hand, the wide variety of ways to implement the algorithm, the success on the initial stage of learning is possible only with the large amount of practical work. At this stage, it is important to gain the experience in simple algorithm implementation and acquire the working skills in one or other tool environment. In the same period, there must be introduced the common programming standards and elements of programming technology and program design techniques. The special attention should be given to the concept of the top-to-down program design and the algorithm decomposition methodology. As was shown by N. Wirth [6], there is some optimum set of algorithms whose realization allows to cover a necessary minimum of knowledge and skills of the initial level in the educational purposes while learning the chosen programming language. The set looks as follows:

- Base data structures and algorithms determined on them.
- Complex data structures: sequences, arrays, records, files.
- Search and sort algorithms on the given data structure.
- Recursive algorithms implementation.
- Dynamic data structures: lists, stacks, turns, trees.
- Program decomposition methods: procedures, functions, modules, objects.
- Algorithms on graphs.
- Interface realization.

The consecutive study of algorithms and their realization in particular language following the principle of increasing complexity of both data structures and algorithms determined on the given structures, allows to solve rather successfully the problem of training in students the basic programming concepts and skills.

4.3 Choosing the Subject Area for Algorithm Implementation

While developing assignments (or tasks) for practical algorithm implementation, a problem arises: what subject area is worthy to be chosen as a base for the assignment generation? Usually for that, the examples from various subjects are used: algebra, geometry, physics, chemistry, biology, mechanics, etc. The main requirement to the assignments is that the subject area should be familiar to the students. A variety of task themes has the advantage that it promotes expansion of student's mental outlook in application of algorithmization methods. However, it requires good correspondence between the students' training background and thematic orientation of the tasks.

Therefore the following programming training method is of interest: some subject area is chosen, its levels on increasing concepts complexity are defined, and the given subject area is offered as a basis for the tasks of programming course with a high degree of correlation on subjects and level of complexity of both the data structures, and algorithms. If the study of the chosen subject area will overtake a little the study of the corresponding programming sections, the

parallel study method of the two disciplines will allow to raise a degree of mastering both programming and chosen subject area. As an example of a subject area it is possible to consider the course of “Numerical methods”. A.P. Yershov [12] pointed out the strong, fundamental connection of the computer science and mathematics concepts:

Mathematics	Computer science
Algebraic system (structure)	Executor (robot, computer)
Carrier	Conditions
Element of the carrier	State of the conditions
Operation	Action changing conditions
Predicate	Question to the conditions
Signature	Instruction set
The protocol. A sequence of operations and predicates with their values + initial element	Behavior. A sequence of actions and questions to conditions with their answers, from the initial state
Predicate-precondition	Task condition
Predicate-postcondition	Task purpose
The available protocol realizing the appropriate predicates of pre- and post- conditions on the ends	The task solution. Behavior leading from the state satisfying the task condition to the state satisfying the task purpose
The program. Sub-recursive set including a set of feasible protocols	The code. The final instructions determining the behavior that leads to the end of each state satisfying the task condition

4.4 Our Suggestion: Mutually Supplementary Course Teaching

Mutual use of different disciplines methods was applied by several authors, for example [13], [14], [15]. While considering the Computational Mathematics or Graph Theory topics, the authors used elements of programming as an illustration of how to realize one or other method. However a crisp interrelation of the given courses was not determined. Our suggestion is that the methods of two courses should be thus interconnected that they supplement and support each other.

Let, for instance, the theme “Complex data structures: arrays” be studied in the course of “Programming” and the theme “Matrix algebra” of “Numerical methods” course have been considering. Then in the study of “Arrays”, students may be given the tasks related to matrix operations: transposition, addition, multiplication, determinant calculation, and so forth. Realizing the tasks in the

form of computer program, students will better mastering the “Matrix algebra”. After that, it would be possible to pass to the next theme of study in “Numerical methods”, for example, “Solving linear systems of equations”. This theme gives the basis to form many assignments for the theme “Dynamic data structures” in “Programming”. In the same manner, links between the other studied sections and courses can be introduced into the teaching and learning process.

5 Project-Based Learning

Most of the existing educational materials on Computational Mathematics provide main theoretical data and sometimes theoretical instructions on how to program a numerical method or algorithm. However, it seems to be inadequate to the end. We believe that the true understanding of a numerical method may be achieved if: (a) a student completes assignments related to a challenging programming project; (b) each project results in practical use of that particular method assigned for the student; (c) the student conducts a set of extensive computational experiments with the program he developed independently; and finally (d) frontal rating of the projects is carried out by the teacher together with the students.

Programming in itself is beneficial for student due to a number of reasons. First, it provides an opportunity to understand and learn a numerical method “from inside”. This is quite different from utilizing ready-made software and significant for any creative professional. Second, it improves student’s computer proficiency, as it requires keen programming. And finally, it develops general analytic and solution seeking performance and implants practical skill to attack and solve computationally oriented problems.

To organize Project-Based Learning while teaching numerical methods to a large audience requires to make programming assignments as varied as possible in terms of the methods’ algorithmic significance rather than their initial data. However, the number of variations on every method is usually limited. In these conditions, finding as many as possible versions of every numerical method becomes a matter of great methodological importance for each teacher.

Organizing PBL means, also, that we should evaluate any laboratory programming project as a single study objective which possesses all the features of a completed software product. Among them are modular structure, convenient interface, efficient utilizing computer resources (memory and time), and possibility to implement a wide plan of computational experiments. This differs definitely from a widely used technique when the students work on one and the same ready-made software when they only enter their initial data and wait passively for a result. The approach we apply makes them perform valuable creative operations, stimulates each student’s competitiveness, prevents cheating and helps to improve overall class performance.

A classic example of how to find as many as possible variant forms of a numerical method is the topic “Elimination and Matrix Inversion Methods”. First of all, the teacher should systemize a set of Gauss and Gauss-Jordan elimination

specific characteristics. They are: (1) direction of elimination of unknowns, (2) mode of access to the matrix entries, (3) mode of updating the active sub-matrix, (4) pivoting strategy etc. Then independence of these characteristics will result in a significant number of different variants of assignments on the same topic being studied.

Over the course on many years, our work is focused on the possible ways of applying PBL to teaching numerical methods in Linear Algebra, Least Squares, Optimal Filtering, Optimal Control, Linear Programming and Nonlinear Optimization. As a result, we recommend that teachers use textbooks, that offer a good choice of various project assignments. The first one [16] contains: Topic 1 - Elimination and Matrix Inversion Methods totalling 26 assignments, Topic 2 - Sparse System Solution totalling 48, Topic 3 - Cholesky Decomposition totalling 40, Topic 4 - Orthogonal Transformations totalling 28, Topic 5 - Simultaneous Least Squares totalling 28, Topic 6 - Sequential Least Squares and Optimal Filtering totalling 25 assignments. The second one [17] contains: Topic 7 - Simplex Method totalling 70, and Topic 8 - Nonlinear Optimization totalling 30 assignments.

5.1 Using the Modern Information Technologies through the PBL

Today, the functioning of a university is impossible without modern tools and technologies of access to information. Universities should pay special attention to the concept of mutually related courses within an educational environment. The development of an individual and self-realization of each student should be considered a priority. Modern information technologies give an opportunity to do so. They allow students to carry out academic tasks with the speed that is an optimum match for each student's talents, skills and character traits.

The mapping out of an educational process in this case consists of the creation of an informational environment.

An information environment assumes the presence of the following components:

1. Interconnected and complimentary academic programs focusing on modern information technologies.
2. A hardware plus software network environment with Internet connections.
3. Computer tools necessary for effective training, including multimedia technologies.
4. Resources for storage, accumulation and usage of data.
5. Organizational actions targeted at providing an effective interaction of all of the participants of the educational process within the modern information environment.

Within a given information environment, the concept of the PBL suggests that students come up with on their own projects to work on rather than use examples offered as a part of the lecture material. Such an approach considerably increases the level of mastering a course.

A great emphasis should be placed on the choice of topics for students' projects, as well as the methods of their realization. It is necessary to take into account both the studied section of the basic academic course, (for example, Numerical Methods) and the knowledge received within a mutually related course (for example, Programming).

The programming course of the Ulyanovsk State University, Ulyanovsk, Russia, uses the following training method based on Project-Based Learning. In the first semester, the students work on their projects, which involves using the studied programming language to program functions and sub-routines. Lab work within these courses is devoted to discussing various alternative methods of programming the same components in the presence of the project advisor.

The topics of assignments are based on the material that has not yet been covered, thus preparing the students for the material that will follow within the Numerical Methods course, for example:

1. Determining a number characteristics: parity, divisibility, and so on.
2. Determining calculation errors.
3. Processing final but unlimited sequences of numbers.
4. Processing one-dimensional and two-dimensional arrays of numbers.
5. Solving equations and systems of the equations.

The program components of every certain task are stored in the component library, i.e. in the module focused on the use of numerical methods.

It is possible to build modules on other subjects, such as computer graphics, graph algorithms, information compression algorithms, information metrics, text processing, sorting and searching, etc.

If a computer class has an Internet connection, then tasks, task specifications, the access to help material, and the demonstrational learning programs become available though the departmental web site, on which all of the educational material is stored. That allows students to use it both in class and at home for independent search. E-mail is a popular means of exchanging information. Teachers use demonstrational and test programs created on the basis of course video materials.

This cycle is completed by the design of the project based on the components created earlier. As a result, teachers reach the basic educational goal, namely the activation of the training process. This process means not only mastering the programming language itself, but also methods of task decomposition, program component design, project assembly and debugging, i.e. all of those elements that are essential for high-level knowledge of the subject.

Moreover, students simultaneously learn concepts and algorithms applied later in the Numerical Methods course that they effectively use for programming.

6 Conclusions

In this paper, leaning upon existing teaching methods and our own experience, we present the Project-Based Learning to prepare students for careers in Computational Science and Engineering.

References

1. Mori, P., Ricci, L.: Computational Science in High School Curricula: The ORESPICS Approach. In: Sloot, P.M.A., Kenneth Tan, C.J., Dongarra, J.J., Hoekstra, A.G. (eds.): Computational Science – ICCS 2002. Lecture Notes in Computer Science, Vol. 2331. Springer-Verlag, Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo (2002) 898–907
2. Anido, L., Santos, J.: An Online Environment Supporting High Quality Education in Computational Science. In: Sloot, P.M.A., Kenneth Tan, C.J., Dongarra, J.J., Hoekstra, A.G. (eds.): Computational Science – ICCS 2002. Lecture Notes in Computer Science, Vol. 2331. Springer-Verlag, Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo (2002) 872–881
3. Jeltsch, R.: CSE Program at ETH Zurich: Are We Doing the Right Thing? In: Sloot, P.M.A., Kenneth Tan, C.J., Dongarra, J.J., Hoekstra, A.G. (eds.): Computational Science – ICCS 2002. Lecture Notes in Computer Science, Vol. 2331. Springer-Verlag, Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo (2002) 863–871
4. Ruede, U.: Computational Engineering Programs at the University of Erlangen-Nuremberg. In: Sloot, P.M.A., Kenneth Tan, C.J., Dongarra, J.J., Hoekstra, A.G. (eds.): Computational Science – ICCS 2002. Lecture Notes in Computer Science, Vol. 2331. Springer-Verlag, Berlin Heidelberg New York Barcelona Hong Kong London Milan Paris Tokyo (2002) 852–857
5. Yip, W.: Generic Skills Development through the Problem-Based Learning and Information Technology. In: Hamza, M.H., Potaturkin, O.I., Shokin, Yu.I. (eds.): Automation, Control, and Information Technology. Proc. of the IASTED International Conference – ACIT 2002. ACTA Press, Anahiem Calgary Zurich (2002) 102–107
6. Wirth, N.: Algorithms and Data Structure; ETH, Zurich (1986)
7. Dijkstra, E. W.: A Discipline of Programming. Prentice Hall, Englewood Cliffs (1988)
8. Turski, W. M.: Computer Programming Methodology. Rheine, London Philadelphia (1978)
9. Kaimin, V.: Course of Informatics: State, Methods and Perspectives. Informatics and Education **6** (1990) [In Russian]
10. Koushnerenko, A.G., Lebedev, G.V.: Programming for Mathematicians. Nauka, Moscow (1999) [In Russian]
11. Dixon, J. R.: Design Engineering: Inventiveness, Analysis and Decision Making. London Sydney (1966)
12. Yershov, A. P.: School Computerization and Mathematical Education. Programming **1** (1990) 3–15 [In Russian]
13. Lipski, W.: Kombinatoryka dla programistów. Warszawa (1982) [In Polish]
14. Ivanova, T. P., Pukhova, G. V.: Computational Mathematics and Programming. Prosveschenie, Moscow (1988) [In Russian]
15. McCracken, D.D., Dorn, W.S.: Numerical Methods and Fortran Programming. John Wiley and Sons, Inc., New York London Sydney (1965)
16. Semoushin, I.V., Kulikov, G.Yu.: Computational Linear Algebra: Collected Assignments for Laboratory Case Studies, Tests and Examinations. Ulyanovsk State University of Technology Publishers, Ulyanovsk (2000)
17. Semoushin, I.V.: A Practical Course in Optimization Methods. Part 1: Linear Programming. Ulyanovsk State University of Technology Publishers, Ulyanovsk (1999)