Reforming Introductory Computer Science Classes to included Inquiry Based Projects

Yenumula B Reddy
Department of Computer Science, Grambling State University, Grambling, LA 71245
ybreddy@gram.edu (http://www.symbolicscience.com)

Abstract - Inquiry based learning is a new educational method that students explore the concepts and problem solving by participating in the complete process of development. The process involves new ideas, questions, suggestions, and criticism during the developmental process. The inquiry based project implementation was introduced in computer science program as part of Center for Mathematical Achievement in Science and Technology to retain and prepare the computer science students for 21st century work force. The initial implementation of the project was introduced in higher level courses for practice and brought the idea in introductory courses. The methodology involves the students in the learning process, improved the learning, and retains the information. The process further helps students to meet the prerequisites for the next level courses.

Keywords: Inquiry based learning; problem based learning; retention; mini-projects; team projects;

1 Introduction

The word “reforming” shakes the community all the times. The reason is that the existing techniques and technology used need to be modified. The people learned and following the traditional techniques need to be changed by incorporating the new techniques as a society or participant group requires. The axiom “Tell me and I forget, show me and I remember, involve me and I understand” is the source to Inquiry-Based Projects (IBP).

The students are changing from computer science major due to fear of programming. This made us thinking to introduce a new method of teaching using problem based learning (PBL) and IBP in introductory classes that help to build skills of programming and problem solving techniques. The team projects at the introductory level were impressed and encouraged by the ABET (Accreditation Board for Engineering and Technology, Inc) team of Computing Accreditation Commission (CAC) visited the computer science program at Grambling State University (GSU). Further, IBP in introductory classes helped us in retaining the students. The PBL at the introductory level is supported by the CMAST (Center for Mathematical Achievement in Science and Technology) program funded by National Science Foundation (NSF) at Grambling State University (GSU).

The traditional methods of teaching the introductory computer science courses use problem solving approach, write algorithm, flow diagram, and then develop a program for compilation. In this process, the instructor discusses the problem solving approach, explains the pseudo code (algorithm), and then enters in programming development. Recently, the methodology was simplified. The simplified method involves the explanation of the problem and then write program (assume the basic elements of the program were taught). The current generation forgets the information after completing the current instruction process because there is no direct involvement of students in developing the program. The instructor gets frustration to teach the same content more than one time. The students get the feeling that the subject is very difficult to learn and change the major instead of wasting time in learning the computer programming. The main reason was the student involvement was missing at every stage (step-by-step) of program development. New ideas, group discussions, develop the simple program or mini-project with complete involvement of students in the program development process.

The PBL is built in the process of observing, questioning, predicting, planning, communication, requirements analysis, and suggesting alternative methods. In the question answer process, the discussions of instructor and students lead to the answer. The process helps to develop concepts, new ideas, discussions, integration of ideas to solution. The PBL focusses on participation and involvement in the learning process unlike traditional methods
concentrated on teacher centered explanation and mastering the subject (not necessary understand everything).

The traditional methods deliver the subject to make students understand the subject with no guarantee of retention. The IBP process clearly shows the complete involvement of each student in the problem understanding and step by step development of the project. Further, the instructor has a clear idea about the capabilities and strength of the student in the developmental process. The process helps the instructor to modify the teaching methodology and introduce the new approaches with student involvement. Further, participation of students throughout the developmental process maximizes the understanding and leads to create small team projects.

In the PBL process, students use the resources beyond the classroom. The resources beyond the classroom involve the students to connect local and world communities. The broad involvement encourages students to ask more questions and participation in problem solving. In this paper, we discuss two examples (due to space limitation) to develop simple program to mini-project.

2 The Real Purpose of Academic Project

It is worth spending time on projects (mini and large) as part of the degree program. The reason is that the student gets both theoretical and hands on experience. Further, there are many skills that cannot be taught in the classroom. Learning and strengthening the skills only possible through practical experience. We identified the following reasons to work on projects.

Working effectively as part of the team: In team work, the participants divide the project into tasks, monitor the tasks by the team leader, integrate, and test these tasks after completion. The weekly or by-weekly meeting helps the progress of the project work.

Interacting with users: Instructor assigns the classroom projects and told to solve it. In the team projects, the team is asked to develop a project through meetings and discussions.

Developing requirements analysis and design: In the classroom students learn specifications, analysis, and design of a project. The concepts can be applied to mini and large projects. Mini projects improve the basic skills whereas large projects involve complex package requirements and software development tools.

Developing prototypes: students normally see the working prototype before they start the project. In a semester time, developing a prototype and implementing the project within a semester is not possible. The mini-project experience helps the team to select large project and can complete on-time.

Writing and oral presentation skills: The projects help students to interact, communicate, written, and oral presentation skills.

Therefore, the project work is not just writing code, but involves team work, selection of problem, time-lines, discussions, and complete development process. Mini-projects are more helpful to strengthen the student’s skills and confidence in developing the projects.

3 First Experimental project

ABET accredited computer science program at GSU identified the following courses as core courses.

<table>
<thead>
<tr>
<th>Course</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 110</td>
<td>Introduction to Computer science I</td>
</tr>
<tr>
<td>CS 120</td>
<td>Introduction to Computer science II</td>
</tr>
<tr>
<td>CS 210</td>
<td>Discrete Structures</td>
</tr>
<tr>
<td>CS 225</td>
<td>Computer Organization and Assembly Language</td>
</tr>
<tr>
<td>CS 235</td>
<td>Data Structures and Algorithms</td>
</tr>
<tr>
<td>CS 300</td>
<td>Computer Science Seminar I</td>
</tr>
<tr>
<td>CS 310</td>
<td>Software Engineering</td>
</tr>
</tbody>
</table>

The remaining courses are built on these courses. The IBP is practiced in CS 110, CS 120, and CS 235 courses. Initially, inquire based projects were tested in CS 310, CS 414 (computer Architecture), CS 424 (Advanced Programming Techniques), and CS 456 (Special Topics in Computer Science). As part of the IBP, the 3 member teams were formed for selection of projects. The selection of the project, feasibility study, and components to include in respective projects were discussed and prepared the time line to complete the projects. The brain storm took at least 5 meetings in two weeks. We created the blackboard approach for open discussions, and started writing ideas to implement the project. The teams completed the requirements analysis and design with group effort and created the algorithms to complete the project. The team participation helped us to complete the project in the scheduled time. The IBP methodology encourages all students particularly students who are not confident to complete a project within the semester time. The
students learned to work on team projects, understood the work, and had good grades.

4 Experiments at Introductory Level courses

The IBP model completed with small teams was brought to introductory level courses CS 110, CS 120, and CS 235, since the retention of the students in computer science is based on these three courses. The retention process starts in the teaching methodology in the first course (CS 110). The process to introduce Java language can be done in two ways. The teaching experience shows that teach the basics through Java program examples or first teach basics and then introduce program writing. Most of the times participants involved more in the first method (starting with an example and slowly introduce language requirements: comments, declarations, expressions, statements, garbage collection, syntax, and semantics). Once the language concepts are clear through examples, the instructor can successfully introduce the small program projects (IBP) through PBL process.

In this document, we discuss two examples due to space limitations. The first example project discusses the problem based learning and second project inquiry based project. In the first program example, we assume that students have the concepts of iterations (for loop and while loop). The development of the first program assumes that the students understand the class concepts and methods with input/output statements.

Example 1: Participating in the first project. Problem based learning

For example, develop a program to add 10 numbers. Assume that stunts have completed the iterations in the classroom. The following steps are involved in the initial level of learning.

**Instructor:** what is the method of adding the first 10 numbers?

Student1: 1+2+3+...+10;  
Student2: add first 10 natural numbers;

**Instructor:** Where do you get these numbers to add?

Student1: read one number at a time and add;  
Student2: read all numbers at one time and add

Student3: Since we know these are first 10 natural numbers we can add them directly without reading (student1 – first step). It saves time.

**Instructor:** All methods are correct. But we need to know the best method among these. If we have to add first 10 natural numbers, use the following method.

```plaintext
n=1; sum =0;  
repeat the following step until n less than or equal to 10  
add n to sum and increase n by 1
```

The simple method is, take the first number and add to the current sum (let current sum be initialized to 0). Increase the number by 1 to get the next number and add to the previous sum. By doing this, you do not need to save all the numbers in memory or type ten numbers on the screen.

In the next step, we need to generalize the method. Therefore, the method must apply to add any number of natural numbers. The method must have the facility of selecting starting number and ending number. We will rewrite the above method as below.

```plaintext
n=starting_Number; m=ending_Number; sum =0;  
repeat the following step until n less than or equal to m  
add n to sum and increase n by 1
```

The above method is general and applies to all similar problems (irrespective of any starting and ending number). The best method means that it should save time, program steps, and computer memory space. In addition, the method must apply for a different set of numbers (may be ten thousand).

It is easy to explain the program steps with team contribution using problem solving techniques. The instructor supervises and helps to complete the implementation of the program on computer. Further, the instructor creates similar problems to strengthen the concepts of problem solving and implementation. Some of the examples are given below that the students will get clarity to understand problem solving process and implementation on a computer.

- Sum of even numbers less than 20
- Sum of odd numbers less than 20
- Product of first 20 numbers
- Product of even numbers less than 20
- Product of odd numbers less than 20

Therefore, an instructor can create many similar problems to get a clear understanding of concepts and solve the problems in the PBL classroom.
Example 2: Inquiry-based project after completion of 2-D arrays (Matrix)

Example 2 assumes the basic knowledge to develop a mini-project. We divide the students into 3 to 5 member groups in the classroom depending upon the class size. Each group develops the same project and participates in the classroom discussions. The problem selected was Magic Square.

Objective:

Students will work cooperatively in a Team and analyze the magic square problem

Rules:

1. Students work in Teams
2. Students discuss strategies to create programmable puzzle

Procedure:

1. The teams were given a start point of the puzzle and the placement of the first 3 numbers
2. All game rules to create the magic square were explained. Each student was encouraged to ask questions while developing the algorithm
3. Starting position of the first number can be changed and other numbers depends upon the change (but make certain that a magic square will be achievable).

Questions considered by the whole group:

1. What are the strategies available to place the numbers on the square with minimum effort?
2. Is there any reason that first number to put in a specific position?
3. Verify the sum of each row, column, and diagonal of the magic square? The sum must be equal in each case.
4. If the first three numbers 1, 2, and 3 are placed as given in the Figure 1, is there another unique solution to the puzzle? Find all possible solutions and follow one to implement that is more suited to implement in Java programming?

Procedure to place the numbers

Initially students were told the procedure to start the 3x3 square and place the first number in the middle of the first row. Students have many questions about the placement. I told them that the placement and staring number can vary. To achieve the solution, we have to select certain strategy and understand the process. Later different methods can be used to achieve the solution. In Java language, indexes start from 0. The index value of the first element of an array is 0. The index value of the first element of a matrix is (0, 0). The matrix must be initialized before you start the process. Test each cell before place the next number. The rules of placement of numbers are explained as below.

- The rows and columns outside the colored part of the square have special meaning. The row above the first row is replication of the last row. The row at the bottom (below the last colored row) is the replication of the first row. Similarly, the right side non-colored column is replication of the first column and left side non-colored column is replication of the last column (rightmost colored column).
- Placement of the next number (2) will be rightmost top corner of the square. Thinking of program design steps, we have to subtract 1 from row and increase column by 1. The process conditions are given below.
  - If the row is less than 0 replace row to maximum (row=2). If the row is greater than 2 (maximum size 2) then set row as 0. If the column index is greater than 2, then set column index as 0. If the column index is less than 0 then set column index as 2. In the current strategy, row will not be more than 2 and column will not be less than 0.
  - If the number is at the rightmost corner, then increase the row by 1 and do not change column (this happens when both row and column crosses boundaries).
  - If the next position is filled by decreasing the row and increasing the column, then row will be increased from current value without changing the column value. For example, after the number 3 was placed the next cell is to be filled by decreasing row and increasing column. In that position, already 1 was placed (occupied). In such situations increase row and no action on column. The number 4 will be placed in the new indexed position.

The procedure to place the numbers in the square was explained more than once to make the students understand. Once they understood the procedure, they completed the magic square on the board. After students understand the problem solving method, I explained the procedure (algorithm) to use in development of the program. The algorithm design completed with many questions related to decreasing row, increasing column, corner cell, filled cell process, and check the row/column boundaries.

Students then tested with any natural number as starting number and larger size of the magic square. Once they understood the process, I explained the
generalization process that we can do any odd number size magic square. The students surfed the literature from Web and showed me the various procedures and strategies to develop the magic square. They found various starting points, various size squares, and placement of numbers in different directions. They implemented the program by moving the next position in different directions. The program is large but using the PBL process students could do this mini-project successfully. There are other mini-projects completed using inquiry based project concepts. The modules will be made available through our tutorials.

CS 235 (Data Structures and algorithms) is a decision level course for computer science students. Change of major after completion of CS 235 (Data Structures and Algorithms) is negligible. Therefore, IBP is very important in introductory courses (CS 110, VS 120, and CS 235).

5 Lessons Learned

Introduction of mini-projects is a challenging task at the introductory level courses. The team formation and dividing the tasks and explain how to divide the project into tasks, responsibilities of each team member and team leader is another problem in the first time. Students understood the process after the first time.

Answering the questions, explain the information sources, and collecting the needed information is another important task in the process. The fusibility study of the project and time-line is another important task. Involving the team members and monitoring each team member to complete the scheduled work on-time is a challenging task.

6 Conclusions

The mini projects at the introductory level make a lot of difference in acquiring the depth of knowledge and learning process. Mini-projects are challenging in introductory courses and participation of students in the project development was impressive. The problem understanding and attempting another new project was improved. Using problem based learning and inquiry based project implementation the computer science faculty members are confident to achieve the projected retention in the CMAST program.

Acknowledgement

The author wishes to express appreciation to Dr. Connie Walton, Provost & Vice President of Academic Affairs, Grambling State University for her continuous support for “Inquiry-based Projects in introductory computer science classes”. Thanks to Dr. Wynn, Dean, Dr. Hubbard, Associate Dean and the computer science faculty Dr. Gassant and Dr. Mesit for their valuable comments and discussions. The author is also thankful to CMAST (Center for Mathematical Achievement in Science and Technology) program funded by National Science Foundation through grant# HRD-1137590 for the support of IBP where PBL.

7 References Cited

3. Problem Based Learning, Michael Wald (Editor-in-Chief), SPECIAL ISSUE, Volume 19 Number 5, Volume 19 Number 5 2003 ISSN 0949-149X.