Teaching Compiler Design Concepts using Case-Based Learning
An Experience Report

Divya Kundra$^1$  Ashish Sureka$^2$

$^1$Deen Dayal Upadhyaya, India - divya1395@iiitd.ac.in

$^2$ABB, India - ashish.sureka@in.abb.com

T4E, 2016, IIT Bombay
# Table of Contents

1. Research Motivation and Aim
   - Case-Based Pedagogy
   - Compiler Course
   - Research Aim

2. Related Work and Research Contributions
   - Literature Survey
   - Research Contributions

3. Solution Approach and Teaching Methodology

4. Performance Evaluation
   - Experimental Setup
   - Survey Based Analysis

5. Conclusion

6. References
# Table of Contents

1. Research Motivation and Aim
   - Case-Based Pedagogy
   - Compiler Course
   - Research Aim

2. Related Work and Research Contributions
   - Literature Survey
   - Research Contributions

3. Solution Approach and Teaching Methodology

4. Performance Evaluation
   - Experimental Setup
   - Survey Based Analysis

5. Conclusion

6. References
Students as the Center of Education Process

Preparing Students for the Professional World

**Case-Based Learning** Learning through solving open-ended real-world problems and scenarios. **Team work** and discussion-oriented

**Cases** can be problem-based, historical in nature, present a model, dilemma-based or demonstrate critical issues in the field [55]

Students apply the theoretical knowledge in solving practical world problems in a supportive environment [14]

**Real world problems** are usually complex, ill-structured, have conflicting choices and can be presented in number of ways to students [24]
Why Compiler Course using Case-Based Teaching

Unexplored - Teaching Compiler Design using Case-Based Teaching

Involves element of programming

Experience of large scale application development

Programming projects are included in the course content
Addition to Case Repository on Compilers

Mini Projects - Lexical and Syntax Analysis

**Aim 1:** Develop cases for teaching essential concepts of compiler design

**Aim 2:** Propose a complete teaching framework that teaches important concepts of compiler design using case-based and project-based learning approaches

**Aim 3:** Investigate the effectiveness of case discussions
# Table of Contents

1. Research Motivation and Aim
   - Case-Based Pedagogy
   - Compiler Course
   - Research Aim

2. Related Work and Research Contributions
   - Literature Survey
   - Research Contributions

3. Solution Approach and Teaching Methodology

4. Performance Evaluation
   - Experimental Setup
   - Survey Based Analysis

5. Conclusion

6. References
### Closely Related Work - 1

<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liu, Guanqun et al.</td>
<td>[33]</td>
</tr>
<tr>
<td>Authors adapt the case-based learning method during the teaching of Delphi language for teaching object oriented concepts</td>
<td></td>
</tr>
<tr>
<td>He, Wu et al.</td>
<td>[17]</td>
</tr>
<tr>
<td>Authors presents a case study to show value of case-based learning in improving the teaching of information security</td>
<td></td>
</tr>
<tr>
<td>Garg, Kirti et al.</td>
<td>[12]</td>
</tr>
<tr>
<td>Authors in [12] have used case studies to teach software engineering.</td>
<td></td>
</tr>
</tbody>
</table>
Closely Related Work - 2

M. Mernik and V. Zumer [36]
Authors have build a tool LISA that supports learning and conceptual understanding of compiler design in an efficient, direct and long lasting manner.

Henry, Tyson R. [18]
[18] suggests that building a compiler for domain specific language (language specially designed for some specific problem) can engage students more than traditional compiler projects.

Ruckert, Martin [43]
texttlx with target processor as the postscript interpreter is a good choice for teaching compilers.
Research Motivation and Aim

Case-Based Pedagogy
Compiler Course
Research Aim

Related Work and Research Contributions

Literature Survey
Research Contributions

Solution Approach and Teaching Methodology

Performance Evaluation

Experimental Setup
Survey Based Analysis

Conclusion

References

Table of Contents

1. Research Motivation and Aim
   - Case-Based Pedagogy
   - Compiler Course
   - Research Aim

2. Related Work and Research Contributions
   - Literature Survey
   - Research Contributions

3. Solution Approach and Teaching Methodology

4. Performance Evaluation
   - Experimental Setup
   - Survey Based Analysis

5. Conclusion

6. References
Unique and Novel Research Contributions

**Contribution 1:** We developed cases for teaching core compiler concepts

**Contribution 2:** We propose and implement a complete teaching framework CPLC which contains learning objectives, case-based and project-based pedagogy and measures students understanding

**Contribution 3:** We evaluate the performance and impact of the proposed case-based pedagogy and demonstrate its effectiveness
Compiler design concepts (CPLC) Model

1. Lectures
   * Compiler design concepts discussed.
   * Theory assignments or tutorials.
   * Programming assignments.
   Didactic teaching Modeling

2. Case Building
   * Identify learning objectives.
   * Cases to be:
     - Interesting and challenging.
     - Stimulus of real world problem.
     - Demonstrate applications of course concepts.
   Exploration

3. Case Allocation
   * Case allocated to teams.
   * 20 mins for case solving.
   * Solving of the case by the team
   * Instructor may give hints.
   Problem Based Learning Articulation Coaching Reflection

4. Case Presentation and Discussion
   * Team presentation (15 mins).
   * Discussion of proposed solution with the class.
   * Interjection by instructor.
   Articulation Inquiry-Based Learning

5. Case Conclusion
   * Solutions submission.
   * Instructor’s feedback.
   * Grading.
   * Project allocation.
   Scaffolding Project based learning

6. Project Presentation
   * Project demo.
   * Instructor’s feedback.
   * Grading.
   Articulation Inquiry Based Learning
Case Study - 1

Case of Spam Detection (Lexical Analysis)

Developers of an upcoming email service - mails.com want to make a spam filter that automatically detects and removes spam. The filter would consist of thousands of pre-defined spam-rules against which the email content will be compared. Anything matching to the spam-rules would categorize to be a spam component. The developers know that as spam filters evolve to better classify spam, the spammers will adapt their writing methods to avoid detection. Thus to build effective rules, the developers of mails.com begin to observe what kind of spam attacks can occur on filters. Example as statistical spam filters begins to learn that word like “offer” mostly occur in spam and starts to think “offer” as spam-rule, spammers began to obfuscate them with punctuation, such as “o.f.f.e.r”. Some of the other attacks are also explained in the case. Observing the attacks discussed in the case and reasoning what other attacks can occur, appropriate tokenization mechanism is to be decided to achieve maximum accuracy of the filter.
Case Study - 1

Technical Challenges

1. Identify various tokenization attacks that can occur on spam filter.
2. Analyze and describe why and how a particular attack can occur.
3. Decide the most promising tokenization techniques that can be proposed for the system.
4. Evaluate the reliability of the proposed tokenization scheme by proving how it will be resilient to the attacks.
GOLEMS is a humanoid robotics lab at GIT. The lab works towards developing robots having human and even super human capabilities. Lab is working on building a physical **human-robot chess**. One side of the chess would have a movable robot arm with sensors providing suitable force to locate, pick, drop and rotate the chess pieces while on other side would be the human playing against the robot. Objectives of the robot is explained in the case. Developers have come up with controlling of the robot using **context-free grammars** which they have called as motion grammar. The production rules of the grammar represent a task decomposition of robotic behavior. The motion grammar enables robots to handle uncertainty in the outcomes of control actions through on-line parsing. The main task is to identify various challenges that will come in design of robot human chessplay system and address those challenges by building the suitable **grammar**. Thus after understanding the requirements and constraints of the system students are required to suggest a promising **motion grammar**.
Case Study - 2

Technical Challenges

1. Identify various requirements of the system to build human-robot chessplay.
2. Identify implicit problems and factors that influence the requirements.
3. Decide and justify the best suitable grammar that can be build which incorporates the requirements of system.
Compiler Design Course - 3rd Year BTech Students

B.Tech students of Deen Dayal Upadhyaya College

48 students in the class

Conduct survey to record student responses

Survey is adapted from a national survey on faculty perceptions of benefits and challenges of case-based instruction [56]

Questions in survey are changed to reflect students perspective on influence of case discussion on different learning subscales [57]

http://dducollegedu.ac.in/
Table of Contents

1 Research Motivation and Aim
   - Case-Based Pedagogy
   - Compiler Course
   - Research Aim

2 Related Work and Research Contributions
   - Literature Survey
   - Research Contributions

3 Solution Approach and Teaching Methodology

4 Performance Evaluation
   - Experimental Setup
   - Survey Based Analysis

5 Conclusion

6 References
Investigating Effectiveness of Case-Based Learning

Survey Design and Responses

Each question offers 4 choices: Strongly Agree, Agree, Disagree and Strongly Disagree

Students choose one of the given option based on their experience about the case discussions

Survey questions categorised under different learning principles along with percentage of students choosing the given option
### Students Response to Survey Questions

<table>
<thead>
<tr>
<th>LP</th>
<th>Question</th>
<th>SA</th>
<th>AG</th>
<th>DG</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning</td>
<td>I felt the use of case-based learning was relevant in learning about course concepts.</td>
<td>14.5</td>
<td>68.7</td>
<td>10.4</td>
<td>6.25</td>
</tr>
<tr>
<td>Learning</td>
<td>The case-based learning allowed for a deeper understanding of course concepts.</td>
<td>10.4</td>
<td>58.3</td>
<td>27.0</td>
<td>4.1</td>
</tr>
<tr>
<td>Learning</td>
<td>The case study will help me to retain different aspects of compilers better.</td>
<td>12.5</td>
<td>52.0</td>
<td>25.0</td>
<td>10.4</td>
</tr>
</tbody>
</table>
## Students Response to Survey Questions

<table>
<thead>
<tr>
<th>LP</th>
<th>Question</th>
<th>SA</th>
<th>AG</th>
<th>DG</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical Thinking</td>
<td>The case study allowed me to view an issue from multiple perspectives.</td>
<td>22.9</td>
<td>64.5</td>
<td>8.3</td>
<td>4.1</td>
</tr>
<tr>
<td>Critical Thinking</td>
<td>The case study was helpful in synthesizing ideas and information presented in course.</td>
<td>12.5</td>
<td>77.0</td>
<td>6.25</td>
<td>4.1</td>
</tr>
<tr>
<td>Critical Thinking</td>
<td>The case study added a lot of realism to class.</td>
<td>14.5</td>
<td>64.5</td>
<td>14.5</td>
<td>6.25</td>
</tr>
</tbody>
</table>
## Students Response to Survey Questions

<table>
<thead>
<tr>
<th>LP</th>
<th>Question</th>
<th>SA</th>
<th>AG</th>
<th>DG</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Engagement</strong></td>
<td>I was more engaged in class during case study.</td>
<td>27.0</td>
<td>62.5</td>
<td>4.1</td>
<td>6.25</td>
</tr>
<tr>
<td><strong>Engagement</strong></td>
<td>The case discussion increased my interests in learning about compilers.</td>
<td>18.75</td>
<td>58.3</td>
<td>14.5</td>
<td>8.3</td>
</tr>
</tbody>
</table>
### Students Response to Survey Questions

<table>
<thead>
<tr>
<th>LP</th>
<th>Question</th>
<th>SA</th>
<th>AG</th>
<th>DG</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication Skills</td>
<td>The case discussion strengthen my communication skills to speak in front of audience.</td>
<td>27.0</td>
<td>58.3</td>
<td>12.5</td>
<td>4.1</td>
</tr>
<tr>
<td>Team Work</td>
<td>The case discussion increased my confidence to work in a team.</td>
<td>22.9</td>
<td>62.5</td>
<td>12.5</td>
<td>2.0</td>
</tr>
</tbody>
</table>
Evaluation Results based on Course Survey by Students

- **Strongly Agree**
- **Agree**
- **Disagree**
- **Strongly Disagree**

<table>
<thead>
<tr>
<th>Category</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Critical Thinking</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Engagement</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Communication Skills</td>
<td>20</td>
<td>80</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Divya Kundra, Ashish Sureka

Teaching Compiler Design Concepts using Case-Based Learning
Percentage of Agree and Disagree Responses

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
<th>Q7</th>
<th>Q8</th>
<th>Q9</th>
<th>Q10</th>
</tr>
</thead>
<tbody>
<tr>
<td>83.2</td>
<td>68.7</td>
<td>64.5</td>
<td>87.4</td>
<td>89.5</td>
<td>79</td>
<td>89.5</td>
<td>77.05</td>
<td>85.3</td>
<td>85.4</td>
</tr>
<tr>
<td>16.65</td>
<td>31.1</td>
<td>35.4</td>
<td>12.4</td>
<td>10.35</td>
<td>20.75</td>
<td>10.35</td>
<td>22.8</td>
<td>16.6</td>
<td>14.5</td>
</tr>
</tbody>
</table>

Divya Kundra, Ashish Sureka
Teaching Compiler Design Concepts using Case-Based Learning
Students have a positive attitude towards the case-based learning methodology

83.2% agreed that case discussions are relevant in learning about compiler concepts

68.7% admitted that case discussions provided them deeper understanding of the concepts of compilers

79% students felt it added realism in the class

Majority of students 89.5% agreed that they were more engaged in class during case discussions
References I

Agnar Aamodt and Enric Plaza.
Case-based reasoning: Foundational issues, methodological variations, and system approaches.

Alexander Aiken.
Cool: A portable project for teaching compiler construction.

Brian A. Altman.

Shawn A Butler.
A client/server case study for software engineering students.
Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. 

Neil Dantam, Pushkar Kolhe, and Mike Stilman. 
The motion grammar for physical human-robot games. 

Akim Demaille, Roland Levillain, and Benoît Perrot. 
A set of tools to teach compiler construction. 
In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE '08*, pages 68–72, New York, NY, USA, 2008. ACM.
References III

Vanessa P Dennen and Kerry J Burner.
The cognitive apprenticeship model in educational practice.

VP Dennen.
Cognitive apprenticeship in educational practice: Research on scaffolding, modeling, mentoring, and coaching as instructional strategies (chapter 31).
retrieved october 9, 2005.

W Diehl, T Grobe, H Lopez, and C Cabral.
Project-based learning: A strategy for teaching and learning.
*Boston: Center for youth development and education*, 1999.

Anne Fuller, Peter Croll, and Limei Di.
A new approach to teaching software risk management with case studies.
Kirti Garg, Ashish Sureka, and Vasudeva Varma.  
A case study on teaching software engineering concepts using a case-based learning environment.

Kirti Garg and Vasudeva Varma.  
A study of the effectiveness of case study approach in software engineering education.  

Vicki L. Golich.  
The abcs of case teaching.  

Audrey Gray.  
Constructivist teaching and learning.  


References VI


References VII


Janet Kolodner.  
*Case-based reasoning.*  
Morgan Kaufmann, 2014.

Janet L. Kolodner.  
An introduction to case-based reasoning.  

When theory meets practice: enriching the cs curriculum through industrial case studies.  

John R Levine, Tony Mason, and Doug Brown.  
*Lex & yacc.*  
Zhaohui Li. Exploring effective approaches in teaching principles of compiler. 2006.


M. Mernik and V. Zumer.
An educational tool for teaching compiler construction.

Megha Mittal and Ashish Sureka.
Process mining software repositories from student projects in an undergraduate software engineering course.

Mark P Mostert.
Challenges of case-based teaching.

Hanspeter Mssenbck.
Compiler construction - the art of niklaus wirth, 2000.
Bringing theory and practice together in engineering classrooms.

Rozilawati Razali and Dzulaiha Aryanee Putri Zainal.
Success factors for using case method in teaching and learning software engineering.

Mary Beth Rosson, John M Carroll, and Con M Rodi.
Case studies for teaching usability engineering.

Martin Ruckert.
Teaching compiler construction and language design: Making the case for unusual compiler projects with postscript as the target language.
References XII

- **Elaine Seymour and Nancy M Hewitt.**

- **Venky Shankararaman.**
  Teaching by case method in computing education.

- **Henry D. Shapiro and M. Dennis Mickunas.**
  A new approach to teaching a first course in compiler construction.

- **Saikrishna Sripada, Y Raghu Reddy, and Ashish Sureka.**
  In support of peer code review and inspection in an undergraduate software engineering course.

- **James Theroux and Clare Kilbane.**
  The real-time case method: A new approach to an old tradition.
References XIII


Elizabeth White, Ranjan Sen, and Nina Stewart.
Hide and show: Using real compiler code for teaching.

Li Xu and Fred G. Martin.
Chirp on crickets: Teaching compilers using an embedded robot controller.

Aman Yadav and Brock E Barry.
Using case-based instruction to increase ethical understanding in engineering:
What do we know? what do we need?

Aman Yadav, Mary Lundeberg, Michael DeSchryver, Kathryn Dirkin, Nancy A Schiller, Kimberly Maier, and Clyde Freeman Herreid.
Teaching science with case studies: A national survey of faculty perceptions of the benefits and challenges of using cases.