CPLC- Case-based and Project-based Learning environment for teaching Compiler design concepts

Farheen Sultana

PRINCIPLE OF COMPILER CONSTRUCTION

Compiler design is a course that discusses ideas usedin construction of programming language compilers. Studentslearn how a program written in high level programming languageand designed for humans understanding is systematically converted into low level assembly language understood by machines.

MOTIVATION AND AIM

Traditional teaching and its disadvantage

A majority of engineering classes involve traditional lecture -based approach in which learning is considered as oriented from teachers to students. The traditional teaching is concerned with teacher being the active controller, having the entire power and responsibility of the environment. The only activity on behalf of students is answering the questions posed by the teacher.

This lecture-based approach is not motivating and does not prepare engineering students well for professional world. Some research scholars and faculties report that use of traditional methods of teaching has led to low level of attendance and retention in engineering disciplines.

Case-based and Project-based Learning

The use of case-based pedagogy can offer solutions to prepare students for the professional world, make education motivating and reduce attrition rates. Case-based learning is different from traditional learning in the manner that it places students as the centre of education process. Students are given importance in what and how they are learning.

Cases can be

- problem-based,
- historical in nature,
- present a model,
- dilemma-based or demonstrate critical issues in the field.

Students apply the theoretical knowledge in solving practical world problems in a supportive environment. Real world problems are usually complex, ill-structured, have conflicting choices and can be presented in number of ways to students

Application of case-based learning is useful in learning about compiler design concepts for the following reasons:

1) **Making learning easier and interesting**: Compiler design course has conceptually difficult topics. It is not easy to teach particularly in small college environment. There are insufficient small grammar examples supported by main textbooks while in reality the grammars for commonly used languages are too complex. Thus, use of contemporary approaches like case-based can enhance the understanding of the course while keeping the class engaging.
2) **Understanding implementation of real-world software**: We develop cases from real world software which use the core concepts of compiler design. We believe students can understand and practice how things actually work in real world.
3) **Skill building**: Through repeated exposure to ambiguous and complex problems in cases, students build confidence and critical thinking. It exposes them to ambiguities and enhances abilities to take timely and effective decisions to unclear and complex problems.
4) **Addition to case repository**: To our best of the knowledge, not much work has been done in teaching compiler.

Compiler design course involves element of programming. Writing a compiler by self can give students experience of large-scale application development. Thus, programming projects needs to be included in the course contents.

The Teachers aims of the work presented are as following:

a) Develop cases for teaching essential concepts of compiler design.

b) Propose a complete teaching framework that teaches important concepts of compiler design using case-based and project-based learning approaches.

c) Investigate the effectiveness of case discussions

Methods Adapted:

Design using case-based teaching methodology. Thus cases developed can be shared and used by other faculty while teaching compiler design course.

This method proposes effective approaches in teaching principles of compiler that includes

- concept mapping,
- problem-based learning (PBL),
- case study and
- e-learning.

Table 1. Teaching plan in *Principles of Compiler*

No.	Chapter	Main contents	Teaching Approaches
1	Introduction	Overview of compiler The process and structure of compiler The exploitation of compiler	Concept mapping E-learning
2	Lexical analysis	Designing method of lexical analysis A simple example of lexical analysis Regular expression (RE) and finite state automata (FSA) The construction from RE to FSA	Concept mapping PBL (case study) E-learning
3	Syntax analysis	Grammar and language Induction and syntax tree Recursive-descent parsing LL(L) parsing Operator priority parsing LR parsing	Concept mapping PBL (case study) E-learning
4	Semantic analysis	Overview Attribute grammar Several intermediate language Expression translation Control sentence translation Array element translation Procedure and function translation Recursive down syntax-directed translation	Concept mapping E-learning
5	Code optimisation and generation of intermediate code	Local optimisation Recycling optimisation Example of optimisation	Concept mapping E-learning
6	Management of run-time storage space	Static storage allocation Simple stack storage allocation Stack realisation of nesting procedure Dynamic stack storage allocation	Concept mapping E-learning
7	Generation of target code	A simple code generator	Concept mapping Case study E-learning
8	Symbol table management and error handling	Symbol table Error handling	Concept mapping E-learning

Table 1 is the detailed plan of the teaching approaches which will be used in this course

- **Using concept mapping**

Concept mapping used in this course is for representingknowledge graphically in a network of interconnectedconcepts. It can be used to: generate ideas, design complex knowledge structures, communicate complex ideas, aidlearning and assess understanding. It can also help toexplain the importance of a particular aspect of a topic sothat students can see how particular pieces of informationfit into the overall schema. It can help students retain a mind map of the information they are studying andunderstand why they are learning it.

I plan to use concept mapping at the beginning of thecourse to show the relationship of this course with the othercourses (see Figure 1).

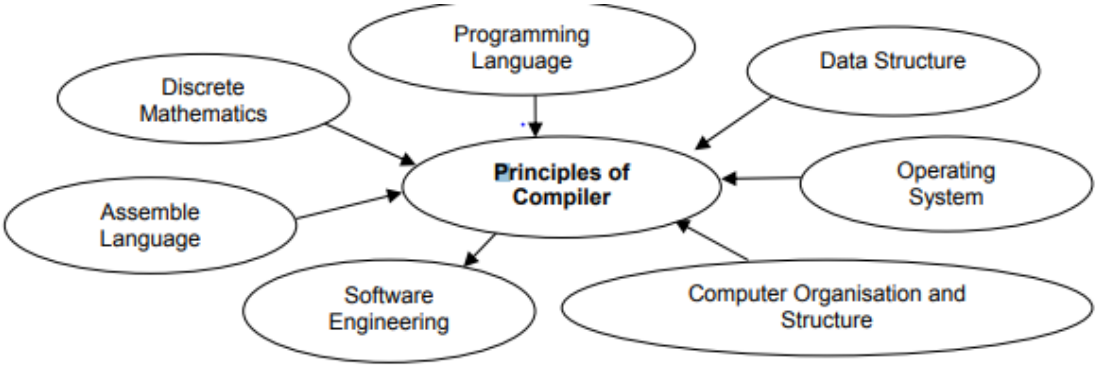


Figure 1. Concept mapping of *Principles of Compiler* with the other courses

This is important to help studentsestablish the whole professional schema. I will also useconcept mapping to show the relationship of the maintopics with this course (see Figure 2).

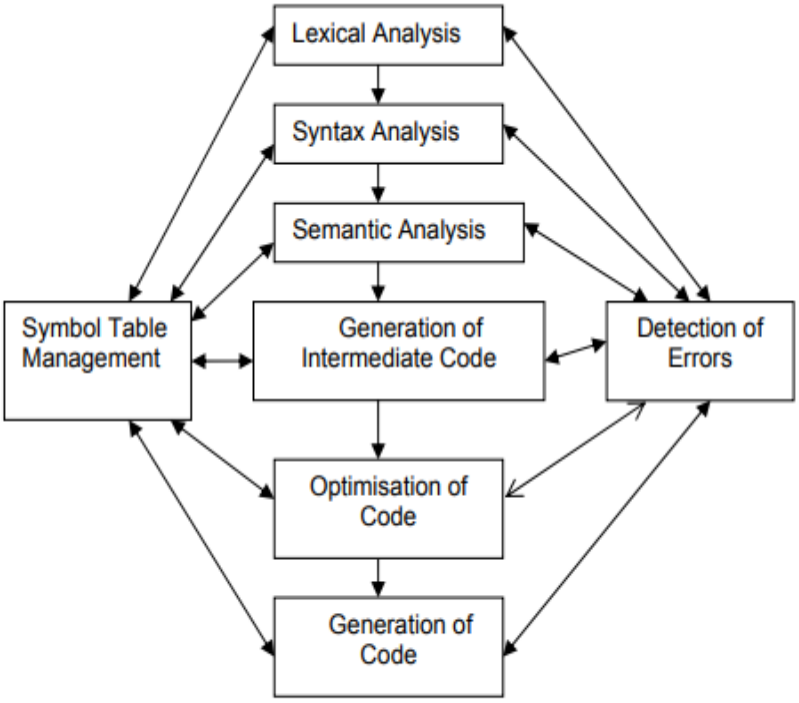


Figure 2. The concept map of main topics

I will use concept mapping in every chapter. At the start ofa chapter, it is used to help students establish theirpreconceptions. It will also be used at the end of a chapterto help students figure out what they have learned and whatthey still do not understand, and thus help the lecture toknow whether or not the students understand the concepts.

Using PBL(Problem-Based Learning)

How to get the students to think and ‘learn to learn’? PBLis an instructional approach which challenges students tolearn by working cooperatively in groups to seek solutions to real problem. PBL used in this course can engagestudents in collaborative learning. Students acquire andapply knowledge in their quest for solutions.

Guided byteachers acting as tutors, they developcritical thinking,problem solving, and collaborative skills. PBL enablesthem to establish relevance between knowledge and realproblems in their learning. This would in turn enhance theircreativity and problem-solving skills. It can increase theirinterest in the course; increase their motivation to learnscience; make them more active in learning; improve theirproblem-solving skills and lifelong learning skills

A sample problem

A. Case of Spam Detection (Lexical Analysis) :

Developers of an upcoming email service -mails.com want to make a spam filter that automaticallydetects and removes spam. The filter would consists ofthousands of pre-defined spam-rules against which theemail content will be compared. Anything matching tothe spam-rules would categorize to be a spam component. The developers know that as spam filters evolvesto 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 observewhat kind of spam attacks can occur on filters.

Exampleas 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 otherattacks are also explained in the case. Observing theattacks discussed in the case and reasoning what otherattacks can occur, appropriate tokenization mechanismis to be decided to achieve maximum accuracy of thefilter.

The challenges for the students in this case are:

a) Identify various tokenization attacks that can occuron spam filter.

b) Analyse and describe why and how a particularattack can occur.

c) Decide the most promising tokenization techniquesthat can be proposed for the system.

d) Evaluate the reliability of the proposed tokenization scheme by proving how it will be resilient tothe attacks.

The teams analysed the case and presented varietyof solutions for the attacks they could identify andsynthesize. Some teams argued that tokenization attackswhich include splitting or modifying key word features(using more of capitalisation or punctuations within theword) are most common and thus proposed solutionsfor them. Some presented obfuscation attacks (changingspelling of spam words to avoid detection) to be a majorspam content and gave solutions for it. A few teamspresented statistical errors such as adding random goodwords to spam or concatenating of small illegitimate

words to form a big permissible word. Teams alsodiscussed about obfuscation of URLs done by encodingor adding unnecessary parenthesis to avoid rule-baseddetection. For data pre-processing different ideas weresuggested. Many of them were to filter out stop wordslike is, an, the and special characters like (), [], performing word stemming and converting all letters into lowercase.

To counter tokenization attacks strategies suggested included scanning the content twice, in the firstscan removal of extra spaces, punctuations within thewords, and in the next scan matching of each tokenagainst bag of spam words (keyword searching). Deterministic Finite Automata's were drawn by students forthe keywords/spams. Some suggested count of punctuations to be an indicator of spam. Idea to use ngrams approach which takes advantage of contextualphrase information (e.g. "buy now") was also proposed.

For statistical errors different solutions presented were:

keeping a count on good words to match against athreshold, weighing the good words against spam words(a significant presence of both can indicate spam) and

keeping a count of location of occurrence of goodwords as some argued that spammers usually insert goodwords in the beginning or at the end. For composite

attacks, ideas mentioned were use of prefix detectionto detect spam by demonstrating the use of REJECT.

construct of YACC as done in the class. For invalid URL,suggestions to do various forms of normalisation of URLwere discussed and for spam present in attachments like
images, discussions were done to process the image toextract set of tokens from properties of image. Thus,the case helped students to contemplate over different tokenization strategies and gain an experience on howcrucial it is to design correct tokenization scheme in thereal-world design of a spam filter.

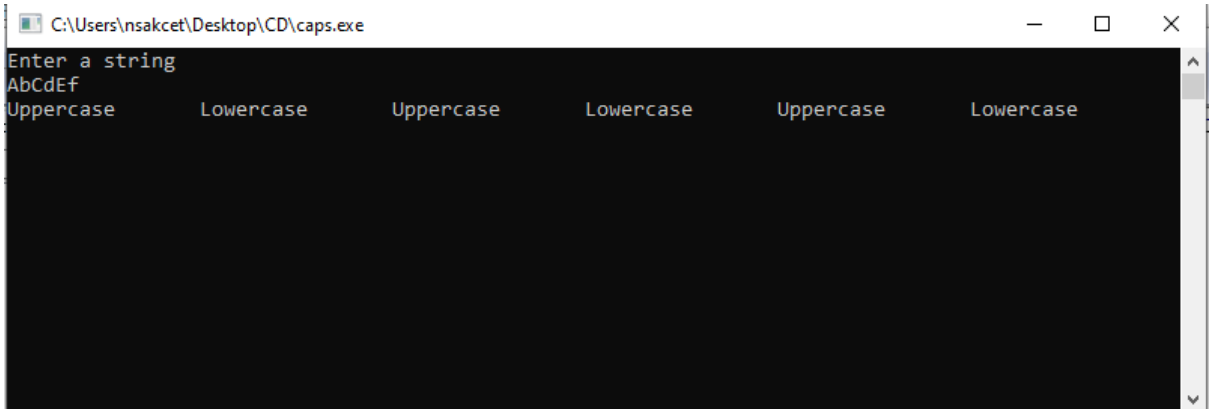
Code:

1. Write a program to identify whether a letter is capital or small

```
%{
#include<stdio.h>
int Upper=0;
int Lower=0;
%}
%%
[A-Z] {printf("Uppercase\t");Upper++;}
[a-z] {printf("Lowercase\t");Lower++;}
%%
int yywrap()
{
return 1;
}
main()
{
printf("Enter a string\n");
yylex();
printf("Uppercase=%d and Lowercase=%d",Upper,Lower);
```

```
}
```

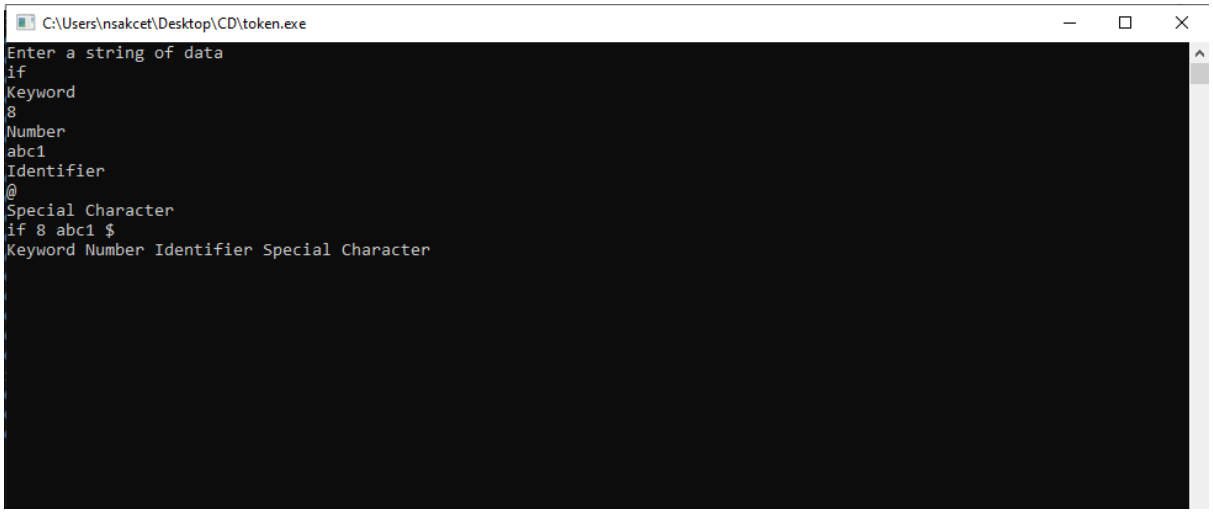
Output:



2. write a program to identify whether a word is keyword,identifier,number or special character.

```
 %{
#include<stdio.h>
%}
%%
"if"|"else"|"while"|"do"|"switch"|"case" {printf("Keyword");}
[a-zA-Z][a-z|0-9]* {printf("Identifier");}
[0-9]* {printf("Number");}
"!"|"@"|"*"|"&"|"^"|"%"|"$"|"#" {printf("Special Character");}
%%
int yywrap()
{
return 1;
}
main()
{
printf("Enter a string of data\n");
yylex();
}
```

Output:



3. Write a program to identify whether a letter is vowel or consonant.

```
%{
#include<stdio.h>
int vowel=0;
int cons=0;
%}
%%
"a"|"e"|"i"|"o"|"u"|"A"|"E"|"I"|"O"|"U" {printf("is a VOWEL");vowel++;}
[a-zA-z] {printf("Is a Consonant");cons++;}
%%
int yywrap()
{
return 1;
}
main()
{
printf("Enter String\n");
yylex();
printf("vowel=%d and Consonent=%d",vowel,cons);
```

}

Output:

B. Case of Human-Robot Chess play (Syntax Analysis)

GOLEMSis a humanoid robotics lab at GeorgiaInstitute of Technology. The lab works towards developing robots having human and even super humancapabilities. One of the tasks of the lab is working onbuilding a physical human-robot chess. One side of thechess would have a movable robot arm with sensorsproviding suitable force to locate, pick, drop and rotatethe chess pieces while on other side would be the humanplaying against the robot. The required objectives ofthe robot are explained in the case. Developers havecome up with controlling of the robot using context-freegrammars which they have called as motion grammar.The production rules of the grammar represent a taskdecomposition of robotic behaviour. The motion grammarenables robots to handle uncertainty in the outcomesof control actions through on-line parsing. The maintask is to identify various challenges that will come indesign of robot human chessplay system and addressthose challenges by building the suitable grammar. Thus, after understanding the requirements and constraints ofthe system students are required to suggest a promisingmotion grammar.

The challenges presented to students in this caseare:

a) Identify various requirements of the system to buildhuman-robot chessplay.

b) Identify implicit problems and factors that influence the requirements.

c) Decide and justify the best suitable grammar thatcan be build which incorporates the requirementsof system.

This case was looked by different perspective by different teams and thus they identified and synthesized differentchallenges.

Each team gave different set of grammars stating different situations they could think can come into humanrobot chess play. While some teams presented a very abstract view of the system in their grammar, few teams did incorporate detailed requirements of the system in their grammar.

First the teams identified the tokens in the system. Someworked with taking tokens as chess states (like checkmate,draw), some worked with robot's movement (like set, release)as tokens while some used sensor's readings (like pressurerelease, pressure set) as tokens.

Student activities:

Group discussion

The students can be divided in groups (6-8 persons). Eachgroup will have a tutorial meeting. Under tutor guidance,students use acquired knowledge and collaborate to find

solutions to the problem. The tutor's role consists of askingquestions, making comments, validating students'solutions, reflect on what they have learned and so on butnot presenting solutions to the problem.

Experimental work

This activity aims at practicing knowledge related to thesolution alternatives to apply the theory knowledge intopractical situation.

Independent study

In order to solve the problem, the students have to doindependent study. They have to search materials and learnnew knowledge and seek solutions to the problem, thus

they develop their self-directed and lifelong learning skills.Skilled learners are more in control of their own learning;create intrinsic motivation rather than extrinsic. They learnin a more active and varied way. They are aware of how tolearn so that they can continue to learn into the future.

Mini-lectures

During the PBL teaching, I will give students some minilectures. The main purposes of giving mini-lectures are tointroduce some basic conceptions, explain some difficult

theories which can not be understood by students andanswer some questions asked by many students and so on.

Using case study

A case study is a student-centred teaching strategybeginning with a story and educating students through thestory. In this strategy, students learn particular concepts,issues or topics through a complete and real-world case.

From an advertisement in the newspaper, you knowthere is a computer company who is willing to paythousands of dollars for the design and construction of alexical and syntax analyser. In this company, a high-level programming language is used for thedevelopment of programs. A compiler for the targetenvironment has been developed. For some reasons, theexistent compiler cannot be used. The manager of thecompany has decided to replace the target environment, so a new compiler must be designed. In order to avoidhaving to design a whole compiler, the managercommissions somebody to design and construct the frontend of the compiler – the lexical and syntax analyser.

In this program, the solution to the problem is not unique.There is some space for students to explore this problem.

Below are a few sample problems that might be given to thestudents.

1. How to describe lexical units formally by using regularexpressions and finite state automata?

2. How to describe the syntax formally by using agrammar? How to analyse and manipulate grammar?

3. How to design and realise a lexical analyser (to designan analyser that reads a source code, checks whether thelexical units in the source code are accepted and returns

the recognised lexical units)?

4. How to design and realise a syntactical analyser (to design an analyser that reads a source code and checks whether the source code is accepted by the grammar)?

E-learning

E-learning is a good approach for teaching and learning.

GitHub is becoming popularas a platform for researchers and scientists to share, update and maintain their dataset as well as code.

Webelieve that sharing our dataset will further facilitateresearch on case-based teaching in computer scienceand in particular on compilers design course and can be used to explore new research problems and hypothesis.Due to limited space in the paper, we briefly describeonly two case-studies, however, we make all the casestudies publicly available through the GitHub repository

Effectiveness of Case-Based Learning

Some effectivecontemporary student-centred teaching approaches such asconcept mapping, PBL, case study, e-learning will beintroduced to this course. These approaches will make thestudents active and responsible for their own learning and thus make the course more interesting, relevant andmotivating. It will improve students' skills forcommunication, problem solving and lifelong learning.