Input: Python || Output: C || Name: PCC Compiler

Evaluation Item	Unacceptable	Marginal	Proficient	Comments
Presentation - Thinking/Inquiry/ Analysis - Innovative Feature (Scored 0-2/[3-4])	The presentation did not convey the project's goals or results. The presentation was not appropriate for the audience. The speaker failed to convey the project's goals and results effectively. Delivery was disorganized/not concise. The innovative feature was omitted from the presentation.	The presentation slightly mistargeted the audience or had trouble with time. The speaker conveyed the project's goals and results. The speaker did not clearly convey the project's goals and results effectively. The delivery could have been more organized/concise. The innovative feature was mentioned but not clearly explained or demonstrated.	The presentation was targeted at an appropriate (technical) audience and was an appropriate length. The speaker conveyed the project's goals and results effectively. Delivery was organized and concise. The innovative feature was clearly explained and demonstrated (where appropriate).	● Goes over some of the technical details of their compiler, such as type checking and how they handled function overloading. ● All members of the project were well versed in their project and were able to explain how they handled some of the technical issues. ● Clearly explains their goal for the project at the beginning of the presentation. Mark: 4/4
Impact - Delivery (Scored 0-2)	The presentation failed to spark interest in many members of the audience.	The presentation made the audience interested about the project and introduced an aspect of the project that was interesting.	The presentation made the audience excited about the project and conveyed a clear sense of its value or potential.	● The motivation behind the project feels somewhat generic. However, they provided a handful of reasons for using such a compiler. Mark: 1/2
Visual Aids (Scored 0-2)	Few visual aids were selected or they were inappropriate for supporting the demo. There were many errors and/or issues with the visual aids.	The selected visual aids were insufficient, too numerous (took over the demo), or inappropriately utilized. There were some errors and/or issues with the visual aids.	The presenter selected and utilized visual aids appropriately to support their presentation. There were no errors/issues with the visual aids.	● Minimal set of diagrams for basic overview of the compiler. Mostly images of code. Mark: 1/2

Q & A (Scored 0-1)	The speaker failed to demonstrate sufficient knowledge in the area of their compiler project with respect to the questions asked.	The speaker demonstrated partial understanding of their compiler project in answering questions.	<mark>The speaker demonstrated a solid understanding of their compiler project in answering questions.</mark>	● Q&A session was well done, the group has a good understanding of their project. Mark: 1/1
Feedback (Scored 0-1)	The student failed to provide complete feedback for all of their peer's presentations.	The student failed to provide complete feedback for some of their peer's presentations.	The student was able to provide useful feedback for all of their peer's presentations.	Yifei: 1/1 Litao: 0.5/1 Jat: 1/1

Additional Notes:
General:
- It was nice to have text broken up with code in the slides and then explained verbally.

CHEN (Jeff) LITAO(1004545842)
- Appears to be reading off of a script, even partially off the screen, as the flow of the presentation is not natural.

SELLATHURAI (Jacob) JATHAVAN(1003403872)
- In the "Primitive types & arithmetic operators" slide, the connection between what was being said and the two code blocks in the slide wasn't clear.
- Some of the concepts weren't introduced clearly before delving into details (e.g. "Constant Folding")

YIN YIFEI (1004910670)
- Was able to clearly explain how the example program works verbally and went over Extended types.

Group Members:

1004545842	CHEN	LITAO	litao.chen@mail.utoronto.ca	chenlita	**Score:**		**7.5/ 10**
1003403872	SELLATHURAI	JATHAVAN	jathavan.sellathurai@mail.utoronto.ca	sellat16	**Score:**		**8/ 10**
1004910670	YIN	YIFEI	yifei.yin@mail.utoronto.ca	yinyife2	**Score:**		**8/ 10**

Student Feedback

Average Score: 4.22/5.0

What they liked?

Use of python type linting, type checking

Good amount of pictures to show examples, and the presentation was clear and concise. Each speaker explained at a good speed.

Informative slides

Very through code examlpes in slide, liked the visual for the compiler architecture

- Good examples and visuals
- Nice and terse explanations, explanations don't run on
- interesting features (function overloading, built in IO)
- good show of optimization

Great visuals. Good explanations (For Ex. Explaining the compiler architecture). Nice to see some additional features also being implemented (Built in IO).

cool idea, loved the function overloading part

everything was well explained and organized

I liked the Python to C and seeing examples.

I liked the overview of the compiler and I liked the support for function overloading

Function overloading -- neat feature!

Very detailed and I liked the compiler a lot too.

Having the three sections for input code, output code, and the output of the output code is a nice addition to be able to see how things run. And also going through the code step by step with code snippets and notes at each step was a nice step to facilitate understanding of what's going on.

Loved the presentation slides, very easy to follow and read along with the presenter. Also the animations we're very appropriate in the examples and added a nice touch, loved the architecture diagram.

They are very detailed when explaining how they implemented their features.

slides an content, very detailed

Nice simple slides. Very easy to follow.

Again, since this presentation was Python-to-C like ours, it was very interesting to see what they did differently--specifically, their compiler architecture was very clear, so we could see that they imported the Symbol Table for the IR instead of embedding it into the IR objects instead.

Their code fragments supporting their slides were well designed

I really liked their slides. They had good presentation skills.

Lot of cool features and extended types, went in depth into the features

They clearly showd their input code, target code and verified the correctness of their target codes. They explained each block clearly.

I really liked that they implemented function type overloading. The technique of hashing the functions is a very smart way to handle it.

Good visuals, easy to follow.

Great examples which explained a lot of aspects of the compiler.

The code snippets were helpful to visualize what was going on.

In-depth and explanations on each component.

Nice project! Well presented.

The presentation flowed nicely and the visuals were pleasing to look at (good use of colours).

I understand the basics of functionality of the compiler from the presentation.

An awesome diagram which illustrates the architecture of their compiler, with very tidy slides comprising of each feature's input code, target code, and expected output.

I like how they showed their compiler's architecture visually with a flow chart
I also like how they brought up errors that are handled.

What they didn't like?

Not enough complex examples

I did not dislike anything.

The pacing is a little fast but I don't blame them given the time.

Thought more slides could have used more visuals

- some parts of presentation where a bit monotone

Instead of an example program, it would've been nice to see a full fledged demo. Showing exactly how the code works all around.

it was overall good

Nothing. I thought it was great

There can be more examples

There wasn't much to dislike about the presentation, the group did a good job overall.

Could be more engaging

Nothing, they covered all bases.

Maybe a little too much white space on certain slides, the text could have just been blown up to be bigger in those sections. Just feels a little weird having the text in those sections just shunted to the left side.

In primitive types and arithmetic operators slide there was a lot of code on the screen, this was different then the other slides, it was very hard to understand.

Nothing, really.

maybe they can improve more eye contact

Small nitpick: The presentation date was March 2021.

I disliked how the way indent/de-indent was handled was only glossed over at a higher level, instead of showing exactly what was done with visuals--this made

it much less clear to me.

I felt their project was kind of generic and as a result their presentation was kind of generic. Maybe it was not possible but I feel that Jathavan presenting from the library was distracting.

nothing, it was good

Their detailed explanation of optimization did not look sufficient enough and i would expect them to talk more about how they are preparing to eliminate dead codes.

I dont dislike anything about the presentation.

Good diagrams, but sometimes not referenced in presentation.

Some slides seemed a bit too crowded.

Slide show could have been a little cleaner in terms of the code snippets placement not running beyond the end and the title not running too far into the left.

N/A

They have the same project idea as us!

The code on the visuals were hard to read because of the font size.

I am a bit confused about the optimization part.

Perhaps elaborate on the purpose of Helper in C component of the compiler architecture within the presentation

I didn't dislike anything about this presentation.

What could be improved?

More examples

A live demo would have been nice.

Seems like following script to the line.

Perhaps add some more visuals in general

- Some presenters were a bit monotone, and seemed to be reading directly off a script
-

It would have been nice to provide a demo. Really cool idea though!

they were quite good

I guess it was fine no improvement needed.

examples of IR AST generated etc would be nice

If I had to improve something, it would be to practice a bit more. It seemed a bit monotone

Can be more interactive

Maybe a live demo would be cool, but it was great!

Besides the one thing noted in dislike, I liked the entire presentation as a whole. Pretty informative and easy to understand.

Definitely not much to change here, loved this presentation. Possibly could have explained the code not he slide i mentioned above, but other than that nothing much.

I think that they're fine

improve more eye contact on it

The slides text could be bigger, especially important for people who have small screens.

Really, I think the only thing the group could have done to improve their performance is to explain exactly what function overloading meant for people that forgot. Asides from that, I really enjoyed everything.

Overall their presentation was good. I feel if they wanted to have an excellent presentation they would have had to implement a more unique feature so their presentation stands out more

it could've been more engaging, but solid presentation

Their helpers in C might affect the performance of their compiler. They could try to move that part to original built in C codes to save some time loading if the input is large.

My feedback for the group is to simply blur their background because there was some activity going on in the background and was a bit distracting.

Would've liked to see a live demo.

Other than a few slides being a bit too crowded, it was great otherwise.

Cleaner slide show would help make everything look a little more professional!

N/A

Nothing, I liked it.

Less reliance on code would help with following the presentation.

The group can explain the optimization part more clearly.

For future improvements, they could elaborate on their choices of optimizations and why implementing those optimizations may be beneficial

I think some parts of the presentation could use a bit more enthusiasm.