

Overview of Project

The implementation project is worth 45% of your final grade, and it is made of three checkpoints, each worth 15%. The project can be done individually or in a group of up to three people. If you decide to work in a group, you are required to sign up on the “Groups” page on CourseLink by 11:59 PM on Feb. 16, 2026. After the due date, you are not allowed to change the members of your groups. If you don’t sign up by the due date, we will assume that you want to work individually for the project so that we can roll out the implementation process in time.

Source Language

C- is a simplified C programming language with some advanced features removed so that we can build a compiler within one semester. Once compiled, C- programs can run on the Target Machine Architecture through the TM (Tiny Machine) simulator. The specification for the C- language is provided in a separate document “C-Specification”. By the end of the semester, you or your group should have a working compiler that takes a C- program as input and produces a TM assembly language program, which you can run on the TM simulator for execution.

Target Machine Architecture

Your compiler will produce assembly language output for the TM simulator, which will be explained later in the lecture notes on “TMSimulator”. The C source code for the TM simulator will also be provided on CourseLink, which once compiled, can be used to execute the assembly language programs generated by your compiler.

Command-line Options

Your compiler will accept several command line options, only one of which can appear for a given invocation of your compiler. Of course, you will build them up incrementally, and only those related to a given checkpoint are required in the related demos and submissions. Here are all the options to be implemented for the project:

- a : perform syntactic analysis and output an abstract syntax tree (.abs)
- s : perform type checking and output symbol tables (.sym)
- c : compile and output TM assembly language code (.tm)

In each case, the command line option instructs the compiler to perform compilation up to the step indicated and output the related results into a file with the same name as the source file, but with the extension of .cm replaced with that indicated for the corresponding option above. Compilation does not need to continue past this point. In the cases where no command line options are given, simply output a message about the usage.

You are free to design the format of your output. However, it should be reasonably complete and understandable, clearly reflecting the internal structures from which it was derived. In the case of -c, the resulting .tm file can be loaded directly into the TM simulator and then gets executed.

Note that outputting the intermediate results for several stages of the compilation process helps us debug a program and show the relevant information during the demos for evaluation. As a result, it will be helpful to show an adequate amount of meaningful information when it is needed, especially when your compiler encounters problems such as core dumps.

Command-line Options

	Topics	Due dates
Checkpoint One	Lexical and Syntactic Analysis (scanner and parser)	Mar. 2, 2026
Checkpoint Two	Semantic Analysis (type checking and symbol table)	Mar. 16, 2026
Checkpoint Three	C- Compiler (code generation)	Apr. 1, 2026

Please refer to separate descriptions for each checkpoint concerning the related submission requirements.