**PROJECT – Overview and Work Plan**

**Overview**

For this semester, the course project will consist in constructing, more or less from scratch (but with the potential help of some tools), an end-to-end interpreted programming language environment. You will design a simple “high-level” PL, scan it, parse it, and interpret it by translating to a simple virtual machine. You will also build some of the aspects of the runtime environment, particularly those pertaining to subprogram call and return.

The main programming language will be C, which you will use for the VM and interpreter, but you may be able to code some of the components in other languages. The project is divided into five phases, which are outlined briefly below. Details will follow, as we get to each of the phases.

This is a group project, and you are expected to work with the same group throughout the project. Groups must have 3-4 members and can include students from both sections. *So, if after Assignment 1 you realized that your cross-section group needs adjustment, this is the time to make the adjustments.*

**Work Plan**

Each phase of the project corresponds to a delivery. We start backwards from the VM, following the order in which material was introduced in the course.

1. For the first delivery, you will design a simple machine language (ML), similar to the pseudo-code language we have examined in Unit 3, and implement its interpreter in C. In addition to the numeric ML will provide a symbolic version of it, as if it were an assembly language.
2. For the second delivery, you will design a simple, relatively high level, PL, describing its lexemes and its grammar using formal methods.
3. The third delivery targets the lexical analyzer for your high-level PL, based on Delivery 2. It will produce the token stream for the next phase.
4. For the fourth delivery, you will build a simple parser, construct a parse tree, and generating the assembly language code, which you will be able to run on your VM after converting to ML.
5. Finally, in the fifth delivery, you will implement runtime structures to execute subprogram call and return.

Details for each delivery will be provided when or before the previous one is due. Please note the tight deadlines and do not procrastinate. For each delivery, one of the team members will be appointed as Team Captain and will have the responsibility of coordinating the work of the team, assuring on-time delivery, and submitting the work for the team (only one submission per team).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Due Date** | **Due Time** | **Course Grade %** | **Handout with Details** |
| Delivery 1 | Virtual Machine, ML Design, AsbL Design | Monday, March 22 | 11:55 PM | 3 | Project-Part1\_VM, ML, AsbL |
| Delivery 2 | High-level PL Design: Lexemes & Grammar | Friday, April 9 | 11:55 PM | 2 | Project-Part2\_HLPL Design |
| Delivery 3 | Lexical Analyzer | Wednesday, April 21 | 11:55 PM | 3 | Project-Part3\_Lexer |
| Delivery 4 | Parser, Parse Tree, & Generator | Monday, May 7 | 11:55 PM | 4 | Project-Part4\_Parser, Tree, Generator |
| Delivery 5 | Subprogram Call & Return | Friday, May 17 | 11:55 PM | 3 | Project-Part5\_Subprograms |
|  |  |  |  | 15 |  |