**Hacettepe University Department of Computer Engineering**

**BBM301: Programming Languages Project**

**Fall 2019**

| Instructors | Dr. Pınar Duygulu Şahin, Dr. Nazlı İkizler Cinbiş |
|---|---|
| Research Assistant | Feyza Nur Kılıçaslan |
| Due Date | 10.12.2019 |

**Description**

Recently, the geographical information systems (GIS) have been used for various type of problems from vehicle tracking to finding the roads. GIS and location intelligence applications can be the foundation for many location-enabled services that rely on analysis, visualization and dissemination of results for collaborative decision making. As a result of growing of mobile smart devices, GIS applications have emerged in these environments.

In this project, suppose that you are developing a programming language to use Geographical Maps and Satellite data (GPS) easily. For example, your PL will be used in programming a navigation system for mobile smart devices and the system shows 3D buildings and structures (such as bridges) and allows users to monitor traffic speeds. You can think roads as graph structures hereby each crossroad becomes a vertex and each road becomes an edge. Your programming language should easily define roads, crossroads and GPS data from satellite and show them on geographical map. You should define built-in functions in your programming language such as ***showonmap(longitude,latitude), searchlocation(address), getroadspeed(road), getlocation(user), showtarget(address)*** for ease of use.

Use your imagination and **develop your own programming language for such GIS applications**.

**Example Program**

For example, a programmer wants to develop a navigation program for mobile smart devices. This program gets the user's current location and guide to user's convenient roads in terms of closeness to target, density of the vehicles, average speed of these vehicles and traffic jam. Programmer wants to do that by calculating a score for each road in other word for each graph edge at specific time intervals (value of time interval can be assigned by users) with respect to road conditions. According to this score users are instructed to the most convenient way to specified target from current position. Programmer wants to take advantage of mobile devices to collaborate with other users who use same navigation program and similar roads for instant location information of them.

**Scope of Your Programming Language**

You should design a programming language to program a geographical information system like in sample navigation program above. You should consider that your programming language should be general for similar problems; for example it should also able to accommodate a vehicle tracking program.

It should include the following base features:

- 3D object definition
- Cross roads
- Road definition
- Showing roads, crossroads on the maps
- GPS data (Longitude and latitude coordinates)
- Score of the roads for convenience
- Collaboration with other users
- Instructing the users

**Part One: Design of the Language**

You will design the syntax of your new programming language. The programming language has to include these ground rules:

- Built-in functions
- Variable names
- Assignment statements
- Arithmetic operations
- Array data type
- Graph data type
- Conditional statements
- Loops
- Function definitions
- Function calls

Your design should include a complete representation of the language in BNF grammar format. The programming language needs to have a name and short explanations for all structures of it.

**Part Two: Lexical Analysis**

In the second part of the project, you are expected to design a lexical analyzer for your PL with lex/flex tools. This lexical analyzer will read an input program written in your language and generate pre-defined language keywords and tokens.

**Part Three: Parser**

For the third part of the project, you are expected to implement a parser for your language using yacc/bison tool. This parser should read an input code and as the output it should state whether the code is written correctly in your language or not. If

the input file is written according to your language's syntax rules, it should print a success message and if it is not, then it should print an error message stating the related line. All the conflicts must be eliminated before implementing a parser. Please make sure that your language is unambiguous clearing all possible ambiguity related definitions.

**About Submit**

The project should be done by groups consisting of at most three people. Your work will be evaluated with regard to the following requirements:

1. Project Report (Soft copy): Project report has to include these:
   - Names and school IDs of the group persons
   - Short descriptions of the defined tokens
   - The complete BNF grammar of your new programming language
   - A list of additional functionality you have included in your language (in addition to the ones provided to you with the base token list)
   - A short tutorial (2-3 pages) on your language, explaining how each language construct can be defined in your language
   - What are challenges of developing a parser for your language? For example, what are the internal ambiguity solutions of yacc that was useful to solve ambiguity of your language, etc.
2. Lex file for your programming language (*.l file)
3. Yacc file for your programming language (*.y file)
4. An example test file which is prepared with your programming language: This test file has to include all the defined structures.
5. A make file which will create a compiler executable with name <language_name>_parser

**Submit:** You have to submit your project with Submit System which will be activated short before the due date. No other type of submission will be allowed. Please be aware that submission of one person from each group is sufficient, not all group members have to submit.

**Submission Format**

- o <studentid>.zip
  - ❖ project(folder)
    - ▪ report.pdf
    - ▪ project.l
    - ▪ project.y
    - ▪ Makefile
  - ▪ test.txt (including code prepared by you that includes a showcase of your language specific features)