**Probal Chandra Dhar  
Advanced Programming Languages**

**Project 3  
User's Manual**

**Setup and Compilation**

1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.
2. The submission includes:

* lex.l
* preced.y
* makefile
* grammar-p3.docx
* userManual-p3.docx
* legal1.txt
* legal2.txt
* legal3.txt
* legal4.txt
* illegal1.txt
* illegal2.txt
* illegal3.txt
* illegal4.txt
* script

1. Environment: This program has been tested in the multi-platform lab and will run there.
2. Compiling. This program includes a Makefile. At the command line in Linux, type make. The program produces an executable entitled compiler

**Running the program:** Be sure the sample programs (legal1.txt, legal2.txt, legal3.txt, legal4.txt, illegal1.txt, illegal2.txt, illegal3.txt, illegal4.txt) are in the same directory as the executable. Issue the command ./compiler < <filename>. Filename is not required to run this program because input is reading from standard input. But if you want to read the input from file please use this command ./compiler < <filename>.

By doing this you can run one program at a time. If you want to run 8 programs at a time you can run the script by issuing the command ./script. Please make sure you’ve to run the Makefile before this command. Type make to run Makefile.

If the syntax of the code in the file match with the provided grammar, then it’ll generate a file same name with the input file but with a .dat extension. For example, if this command

./compiler legal1.txt

given after the make command then it’ll generate output like this:

Successfully compiled.

If there is any error in the input file or in the command you gave to the compiler, it’ll generate an error message like this:

error is syntax error

error is error

Script is a bash script that will run the program 8 times with different sample program I’ve added in the folder.

**User input:** no user interaction with the program is required if you use input redirection < in the commandline. If you don’t want to read input from a file you can just use this command ./compiler and then type any statement you want to compile.

**Output:** all output goes to the console. Output will be the similar to this:

Successfully compiled.

Successfully compiled.

Successfully compiled.

Successfully compiled.

error is syntax error

error is error

error is syntax error

error is error

error is syntax error

error is error

error is syntax error

error is error