User manual for compiler

**Install**: Mars4\_5.jar(<http://courses.missouristate.edu/kenvollmar/mars/>)

JDK(to run the jar package)

**Run**:

Click the compiler.exe to open this compiler.

When you open this software, you will see a manual in which you can make some choices.

1. **open a text file for writing code**

system will open the text file LPAS.txt, you can write Lpascal in this file and store them.

1. **compile the code**

compiler will compile the Lpascal language into mips code.

1. **show the mips code**

system will open the text file codefile.txt which records the mips code that you just generate.

1. **run code in Mars**

you should follow what the compiler says that “there is a Mars4.5 in the file, you can open it and copy the mips code(from codefile.txt) to it”

1. **close the compiler**

if you input ‘5’, the compiler will be closed.

code\_generation.exe: click it to open the compiler.

LPAS.txt: you can write the Lpascal code in this file.

codefile.txt: this file record the mips code the compiler just generated.

Mars4\_5.jar: this software can run the mips code.

**Little Pascal grammar**

All of codes should begin from a Program, and the following code should look like this PROGTOK IDTOK ‘(‘ ‘)’ CONSTPART VARPART BEGTOK stat morestats ENDTOK IDTOK'.'

You can see details in the parserBNF.docx which record the little pascal grammar.

The limitation is that all of the

**Code Reference:**

Program dumb (); (\*program dumb \*)

var

n : integer; reverse: integer; temp : integer;

begin

writeln("enter a number to check if it is a palindrome or not");

read(n);

temp :=n;

while(temp>0) do

begin

reverse :=reverse\*10;

reverse :=reverse+ temp mod 10;

temp := temp/10

end;

if (n=reverse) then writeln("yes!")

end. (\*hahahahahahahaha\*)

(Don’t input a semicolon before a end token!)

**Error:**

Error("it'an illegal token!")-- scanner scan a alphabet after a number

Error("This variable has not been declared!")-- miss a declaration of this variable

Error("Same variable has been declared in this block! ")—can’t find the variable in the symbol table

Error("types mismatch in assignment")—the type of left operand doesn’t match the right operand.

Error("Do not allow read into const value.")—read a value into the const variable

Error("Do not allow read boolean.")—read a Boolean variable

Error("Don't allow write the boolean type")—try to write a boolean variable

Error("type mismatch in expressprime")-- the type of left operand doesn’t match the right operand in expressprime.

Error("type mismatch in termprime")-- the type of left operand doesn’t match the right operand in termprime.

error.setErrRout()—in each beginning of the parser part, assign the routine to the Error routine.

error.setErrTok(get<1>(Token))-when some wrong occur in some tokens, it will record the wrong token.

**Limitation**:

Error report: If the program catches some errors but doesn’t stop because of some reasons, the Error routine and Error token will be set a wrong value.

This compiler can’t report the line of the Error, I will improve it in the future work.

You should close the compiler after you run the execution, otherwise the symbol table will insert new symbols continuously. If you want to write new code, you can open it again.

Variables can’t truncated to 8 characters and case sensitive(Ex, you should input all of the keyword with lowercase letter).