

Accepting types

Datatype : int, double String char

Op : +, *, /, -, %

Var : !Keyword && ^[_a-zA-Z][0-9a-zA-Z]*

Keyword :

"abstract", "continue", "for", "new", "switch", "assert", "default", "goto", "package", "synchronized", "true", "false", "null", "boolean", "do", "if", "private", "this", "break", "double", "implements", "protected", "throw", "byte", "else", "import", "public", "throws", "case", "enum", "instanceof", "return", "transient", "catch", "extends", "int", "short", "try", "char", "final", "interface", "static", "void", "class", "finally", "long", "strictfp", "volatile", "const", "float", "native", "super", "while", "String", "float", "long"

Val : Int

|Char
|String
|Double
|not initialized
|Invalid

Int : \\d+ (positive values only)

Char : '[\\w\\d]'

Double : \\d+\\.\\d+

String : "[\\w\\d]*\\" (Accepts only single word)
Else Invalid value

Grammar

Code : start STAT stop

STAT : DECL
|ASSIGN

DECL : Datatype Var Eq ASSIGNVAL;
|Datatype Var ;

ASSIGN : Var Eq Val
|Var Eq Var
|Var Eq EXPRSN

ASSIGNVAL : Var
|Val

EXPRS : Val Op Val
|Var Op Val
|Val Op Var
|Var Op Var

Valid Assignments

Var_datatype	Accepted Value data type
int	int,char
char	char,int
double	int,char,double
String	String

Expression validation

For op = +

+	int	char	String	double
int	i,c,d	i,c,d	String	double
char	i,c,d	i,c,d	String	double
String	String	String	String	String
double	double	double	String	double

Other operands

String is invalid for op other than + ;

For op = " - "

If value < 0 char throws error

op	int	char	double
int	i,c,d	i,c,d	double
char	i,c,d	i,c,d	double
double	double	double	double

