**COVER PAGE**

# **CS323 Programming Assignments**

Fill out all entries 1 - 7. If not, there will be deductions!

1. Names [ 1. Justin Stitt ]

2. Assignment Number [ 3 ]

3. Due Date [11 December 2021 ]

4. Submission Date [11 December 2021]

5. Executable File name [Parser.exe]

6. Names of the testcase and output files (output files in )

test1.rat -> test1.out

test2.rat -> test2.out

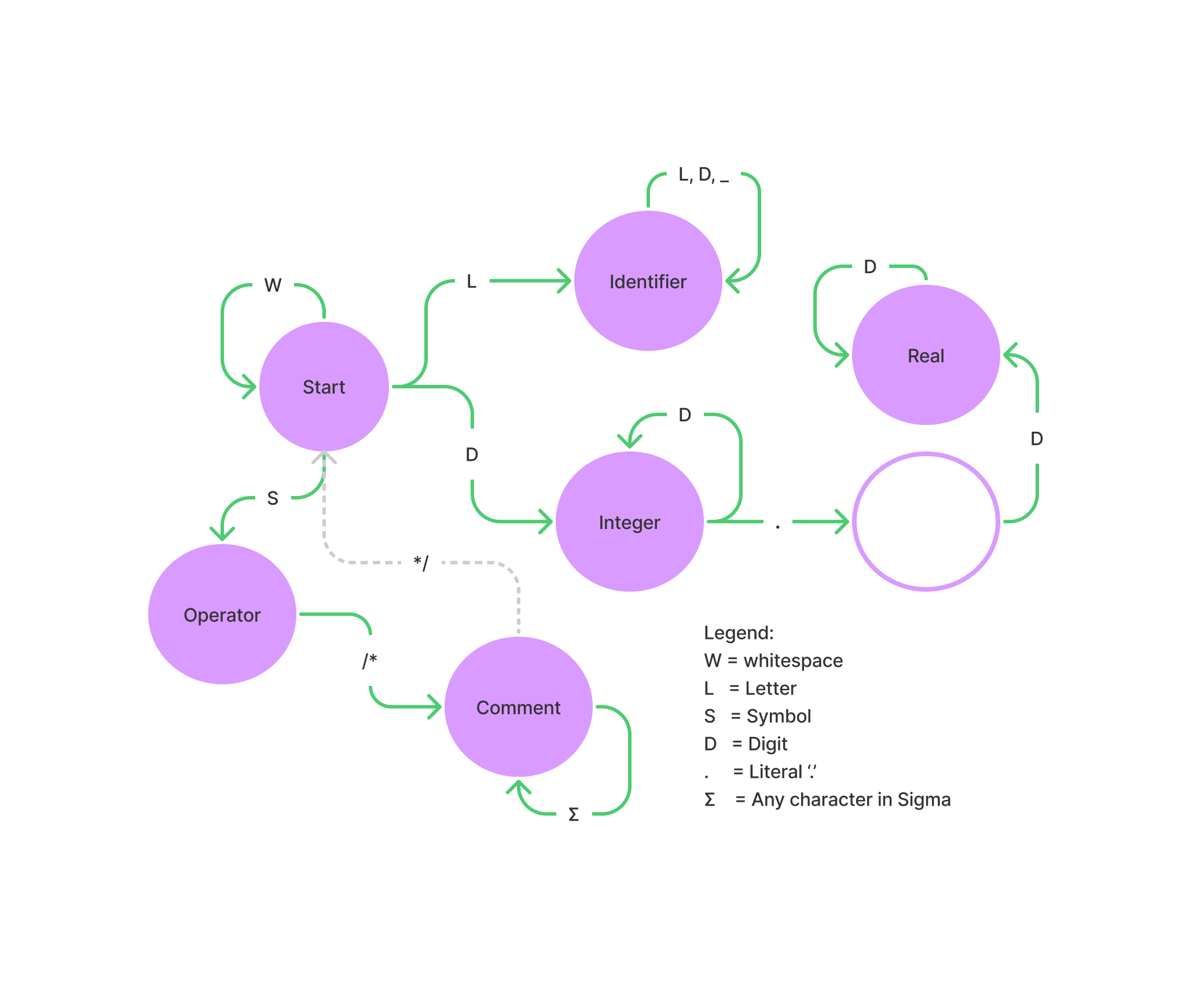
test3.rat -> test3.out

7. Operating System [WINDOWS]

Comments and Grade:

CPSC 323: Assignment 3

Designing and implementing a Symbol Table as well as Generating Assembly Code for A Simplified Rat21F



# 

# Problem Statement

Write and design a Symbol Table to hold variables, their types, and position in memory starting from *7000.* We must also output appropriate assembly instructions as defined by a simplified version of the Rat21F language.

# How to Use the Assembly Code Generator

1. Download files
2. Run Parser.exe
3. Type in the filename of a file in the same folder as the .exe (example: "test1.rat")

# Design

(Source code in './src' folder)

Implemented a Symbol Table class that uses a map to connect identifiers to their appropriate memory location. I also use another map to hold their types (integer, boolean).

I designed and implemented the following constructs/functions for Assignment 3:

* Instruction class to store 3-address-code formatted information (3ac).
* SymbolTable class
* SymbolTable::add
* SymbolTable::find
* SymbolTable::print
* SymbolTable::set\_batch\_id\_types
* SymbolTable::set\_id\_to\_type
* parser::jumpstack
* parser::back\_patch
* parser::gen\_instr
* parser::print\_instr
* … as well as other auxilliary functions

My assignment utilizes an OOP (object-oriented programming) approach that helps me compartmentalize all the moving parts within assignment #3. My lexer, parser, symbol\_table, and instruction classes all work hand-in-hand.

*It should be noted that my symbol table does not fill up "linearly" because I am using a map (binary search tree). Therefore you may see that the order in which some of the identifiers appear in the symbol table do not match the order in which they were declared. This is NOT a functional limitation and is by design (it's faster).*

# Limitations

* If parsing fails, it is likely that assembly code generation will also fail. Correct Rat21F code must be given as input for any chance at correct assembly code output.

# Shortcomings

None