Project Phase2: Semantic Analysis and Symbol Table
Due Date: Wednesday February 8th, 2023 (11:59 pm)

---

1 Star Language

In this phase, you will reuse the code that have been submitted in phase-1 to perform type checking for **Star** language.

a) Declaration Checking: For this part, you need to process the declaration.

- A variable cannot be used if it is not previously declared.
- No two variables can share a name within their declared scope.

b) Type System: Star specification also describes rules for statements' and expressions' type checking.

- The left and right side of an assignment statement must have the same type.
- Arithmetic operators (+,-,*,/) operate on operands of type integer or float only. The result should be of same type of the two operands, if one operand is integer and the other is float the result should be of type float.
- Relational operator, the two operand must have the same type and the result is boolean (true or false).
- Logic operator (&&,||,!) only operates on boolean expression and the result is boolean (true or false).
- The expression/condition in (if and while) statements must be of type boolean.

for simplicity assume the Star language have only +,* ,&& and ==, <> operators

c) Error Handling
During parsing, if one of the above rules (a declaration or a type rule) is violated, print an error message that explains in details the violation (along with associated line number) and continue processing the input. Your compiler should detect and show all semantic errors that occur before the first syntax error. In case of syntax error, your compiler should show a message with the line number of the first encountered syntax error then exit the system.

d) Input/Output Samples:

```
START
int X12
float ABC1
X12 = 2
while(X12 == 0)
{
   ABC1 = 4.5
   if( true )
   { print("Inside␣IF␣inside␣Loop") }
   end
}
print ("Hello␣..␣")
END
```

myP.txt

```
START
int X12
X12 = 2
while(X12 == 0)
{
   ABC1 = 4
   if( true )
   { print("Inside␣IF␣inside␣Loop") }
   end
}
print ("Hello␣..␣")
END
```

myP1.txt

```
START
int X12
X12 = 2.5
while(X12 == 0 )
{
   ABC1 = 4
   if( true && 1)
   { print("Inside␣IF␣inside␣Loop") }
   end
}
print ("Hello␣..␣")
END
```

myP2.txt

```
START
int X12
X12 = 2
while(X12 == 0 )
{
    float X12
    X12 = read()
    if( true && X12 <> 1)
    { print("Inside␣IF␣inside␣Loop") }
    end
}
print ("Hello␣..␣")
END
```

myP3.txt

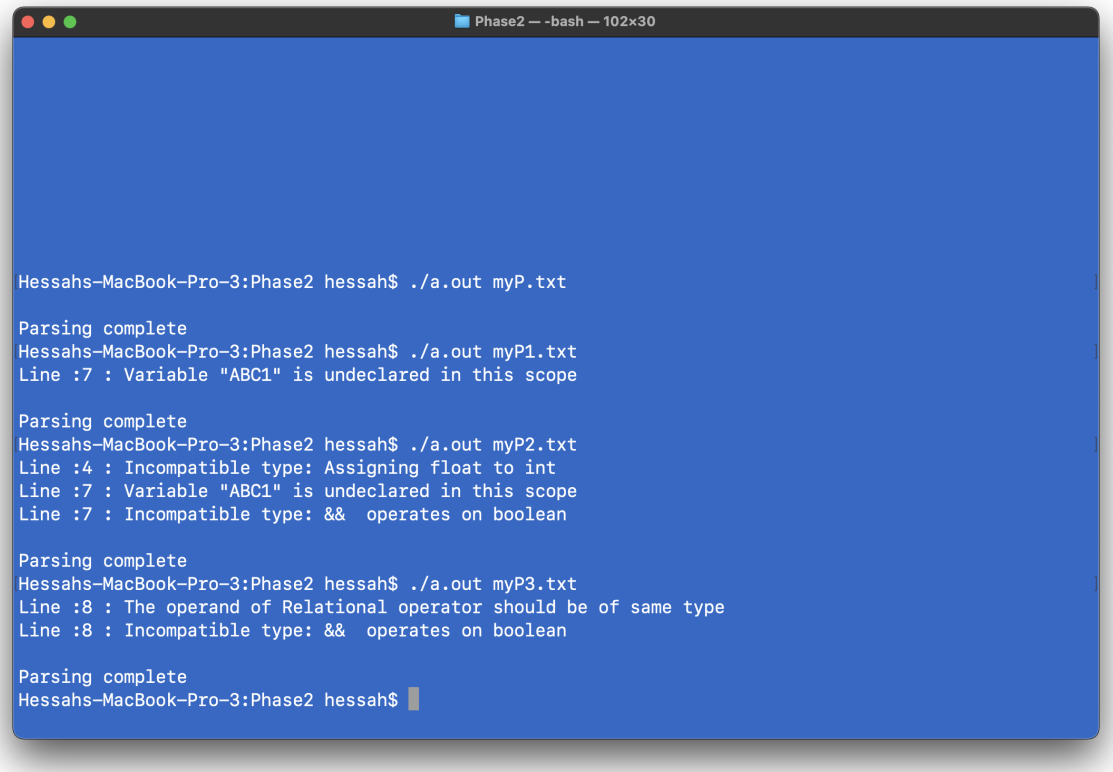

Figure 1: Parsing myPX file

2 Deliverable and rules

Write the Lex and Yacc file for Star language as described above. Once you built your parser, you should be able to parse Star programs.
You must deliver one zip including:

1. A soft copy of a lex file "phase2.l"
2. A soft copy of a YACC file "phase2.y"
3. A soft copy of header file "phase2.h"

Due Date: Wednesday February 8th, 2023 (11:59 pm)