Code Generation

Compiler Design (COMP 442/6421)
Assignment 5

Name: Garvit Kataria
Student Id: 40155647

Analysis

Implementation

1.	**Memory allocation**	marks
1.1	Allocate memory for basic types (integer, float).	1.0
1.2	Allocate memory for arrays of basic types.	0.5
1.3	Allocate memory for objects.	0.5
1.4	Allocate memory for arrays of objects.	0.5

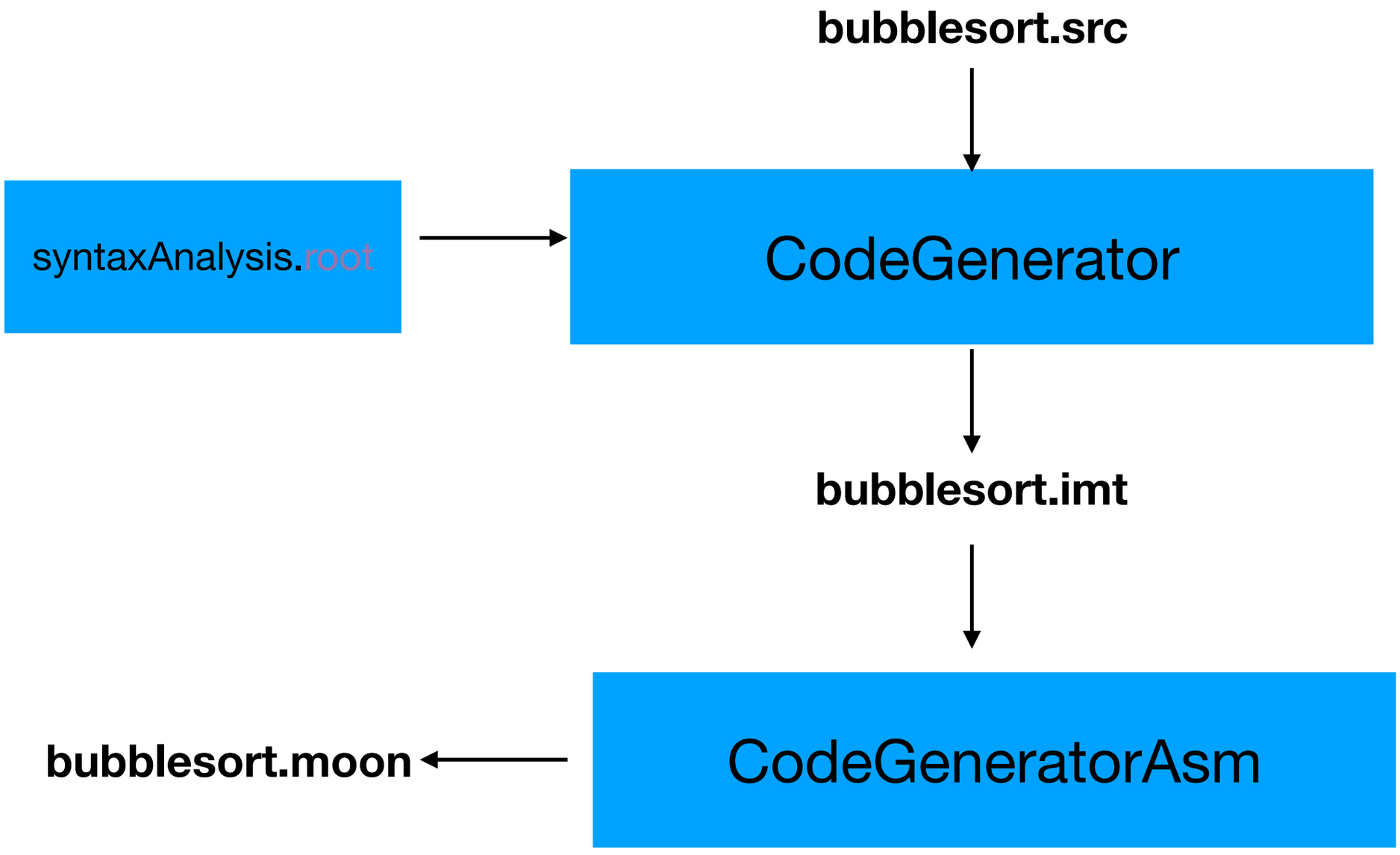
2.	**Functions**	marks
2.1	Branch to a function's code block, execute the code block, branch back to the calling function.	1.0
2.2	Pass parameters as local values to the function's code block.	1.0
2.3	Upon execution of a return statement, pass the return value back to the calling function.	1.0
2.4	Call to member functions that can use their object's data members.	1.0

3.	**Statements**	marks
3.1	Assignment statement: assignment of the resulting value of an expression to a variable, independently of what is the expression to the right of the assignment operator.	1.5
3.2	Conditional statement: implementation of a branching mechanism.	1.0
3.3	Loop statement: implementation of a branching mechanism.	1.0
3.4	Input/output statement: execution of a keyboard input statement should result in the user being prompted for a value from the keyboard by the Moon program and assign this value to the parameter passed to the input statement. Execution of a console output statement should print to the Moon console the result of evaluating the expression passed as a parameter to the output statement.	2.0

4.	**Aggregate data elements access**	marks
4.1.	For arrays of basic types (integer and float), access to an array's elements.	1.0
4.2.	For arrays of objects, access to an array's element's data members.	1.0
4.3.	For objects, access to members of basic types.	1.0
4.4.	For objects, access to members of array or object types.	1.0

5.	**Expressions**	marks
5.1.	Computing the value of an entire complex expression.	2.0
5.2.	Expression involving an array factor whose indexes are themselves expressions.	1.0
5.3.	Expression involving an object factor referring to object members.	1.0

Design


bubblesort.src
syntaxAnalysis.root
CodeGenerator
bubblesort.imt
bubblesort.moon
CodeGeneratorAsm

Design

- **CodeGenerator converts src files like bubblesort.src to an intermediary file i.e. bubblesort.imt which then goes to CodeGeneratorAsm module which converts it to moon file i.e. bubblesort.moon**

ifStat_3.src

```
func main() -> void {
    let x: integer;
    x=4;
    if (x < 5) then {
        write(x);
    } else;
}
$
```

src file

ifStat_3.imt

```
start functionDef# main :void
start generateStatementBlockCode
varDeclare# integer x
assignStatement# x = 4
start generateIfStatementCode#
start generateRelExprCode
condition# x < 5
end generateRelExprCode
start block 1
start generateStatementBlockCode
writeStatement# write x
end generateStatementBlockCode
end block 1
start block 2
start generateStatementBlockCode
end generateStatementBlockCode
end block 2
end generateIfStatementCode#
end generateStatementBlockCode
end functionDef# main
```

imt file

ifStat_3.moon

```
entry
addi   r14,r0,topaddr  % Set stack pointer
main
addi r1,r0,4
sw mainx(r0),r1
lw r1,mainx(r0)
clti r2,r1,5
bz r2,block2
lw r1,x(r0)
putc r1
j endif1
block2
endif1
hlt
mainx      res 4
buf      res 40
```

moon file

Memory allocation

```
| table: global
================================================================================
| class     | POLYNOMIAL|     | 0
|    ============================================================================
|     | table: POLYNOMIAL
|    ============================================================================
|     | inherit     | none
| function   | evaluate| (float,): float  | public
     | 0
|    ============================================================================
|     | table: evaluate
|    ============================================================================
================================================================================
| class     | LINEAR|     | 16
|    ============================================================================
|     | table: LINEAR
|    ============================================================================
|     | inherit     | POLYNOMIAL
| function   | evaluate| (float,): float  | public
     | 8
|    ============================================================================
|     | table: evaluate
|    ============================================================================
|     | local      | result      | float      | 8
================================================================================
| function   | build| (float, float,): LINEAR  | public
     | 0
|    ============================================================================
|     | table: build
|    ============================================================================
|     | local      | new_function      | LINEAR      | 16
```

```
================================================================================
| class     | QUADRATIC|     | 24
|    ============================================================================
|     | table: QUADRATIC
|    ============================================================================
|     | inherit     | POLYNOMIAL
| function   | evaluate| (float,): float  | public
     | 8
|    ============================================================================
|     | table: evaluate
|    ============================================================================
|     | local      | result      | float      | 8
================================================================================
| function   | build1| (float, float, float,): QUADRATIC  | public
     | 0
|    ============================================================================
|     | table: build1
|    ============================================================================
|     | local      | new_function      | QUADRATIC      | 24
================================================================================
| function   | main     | (): void      | 4
|    ============================================================================
|     | table: main
|    ============================================================================
|     | local      | counter      | integer      | 4
|     | local      | f2      | QUADRATIC      | 24
|     | local      | f1      | LINEAR      | 16
================================================================================
```

- **Memory allocation of all the classes and variables is done by the SymbolTableGenerator.**

Use Of Tools

No use of 3rd party tools for this assignment.