Sonoma State University
Computer Science Department
CS460 – Fall 2023 – Watts

Semester Project - Part 3 (aka Project 3)

Project Grammar: http://watts.cs.sonoma.edu/cs460f21/ProjectGrammar.pdf

For this part of the Project you will be generating C++ code for each of the PL460 input files.

Specifications

1. There is a starting version of the CodeGenerator class available in the directory Project3Framework in the course pickup folder. This folder contains files for a fully functional Project 2 Syntactical Analyzer. You may rename and complete these files or (better yet!) you can use your Project 2 files.

2. Calls to the Code Generator should be made from your Syntactical Analyzer.

3. For each .pl460 input file, your program should generate a .cpp file with the same name, just a different extension. For example, if the input file is P3Test1.pl460, the output file should be P3Test1.cpp.

4. The Object class (also in the Project3Framework directory) should be included in your generated C++ files so that you can use the generic type Object in your generated code. Your C++ program will need to include "Object.h" and will need to be linked with Object.o.

5. The generated code should compile with no errors and should produce essentially the same output as the original PL460 code. An example is at the end of this document. Output differences may appear due to default precisions and some symbols.

6. All of your source code should be placed in a folder called lastnameP3. The folder should contain the code required to generate an executable called P3.out.

7. Your folder should also contain a makefile. The first target of your makefile should be P3.out.

8. Your folder should also contain a file called README.txt that describes what your project submission does and what if does not do.

Date due: Thursday, 14 December 2023 at 11:59 pm.

To turn in: A tarred and zipped directory containing source files (headers and implementations), your test .pl460 files, a makefile and README.txt. Your directory should be called *yourlastname*P3 and your tarred and zipped file should be called *yourlastname*P3.tgz. Submit your .tgz file by copying it to ~tiawatts/cs460drop.

Sample PL460 Input (P3-example.pl460):

```
;; Project 3 Example

(define (listop_ex1)
     (cons (car '(a b c)) (cdr '(d e f)))
)

(define (listop_ex2)
     (cons (cadr '(a b c)) (cddr '(d e f)))
)

(define (main)
     (display (listop_ex1)) (newline)
     (display (listop_ex2)) (newline)
)
(main)
```

Possible Generated Code:

```
// Autogenerated PL460 to C++ Code
// File: P3-example.cpp
#include <iostream>
#include "Object.h"
using namespace std;

Object listop_ex1 ()
{
     Object __RetVal;
     __RetVal = listop ("cons", listop ("car", Object("(a b c )") ),
                                listop ("cdr", Object("(d e f )") ));
     return __RetVal;
}

Object listop_ex2 ()
{
     Object __RetVal;
     __RetVal = listop ("cons", listop ("cadr", Object("(a b c )") ),
                                listop ("cddr", Object("(d e f )") ));
     return __RetVal;
}

int main ()
{
     Object __RetVal;
     cout << listop_ex1();
     cout << endl;
     cout << listop_ex2();
     cout << endl;
     return 0;
}
```

Expected output:

```
(a e f)
(b f)
```