Final Exam

EECS665 - Compiler Construction
2019, Spring

Name: ____________________ Student ID: ______________

Do Not Open Until Instructed!

Before the Exam starts:

- Read all of the instructions on this page
- Write your name and student ID on this page
- Retrieve your page of notes and writing materials
- Put all other materials away and silence your devices

After the Exam starts:

- Write your student ID (**not** your name) on all subsequent pages
- If you feel a question is wrong or impossible, notify course staff.
- Announcements / corrections will appear on the projector
- Turn in all your related paper when finished, including:
  - your notes pages
  - reference pages
  - the exam itself
  - scratch paper
- You may leave when done (no new material will be presented).
- Work quickly, move on if you are stuck.

Feel free to draw something inoffensive in the box below to pass the time

Total Questions: 12
Time Limit: 135 minutes
Pages:

- 12 pages total

Score: ______ / 175 pts

QUESTION 1 (15 POINTS)

Student ID: ____________________

Consider the language Z recognized by the following ambiguous grammar:

$A \longrightarrow A$ **zip** A
$A \longrightarrow A$ **zap** A
$A \longrightarrow A$ **zop** A
$A \longrightarrow$ **num**
(note that A is the start symbol of the grammar)

Write a grammar for Z with the following properties:

- The grammar is not ambiguous
- **zip** has highest precendence
- **zap** has next precendence
- **zop** has lowest precendence
- **zip** is left-associative
- **zap** is right-associative
- **zop** is left-associative

Question 2 (15 Points)

Student ID: ____________

Draw out a parse tree for the following token string that obeys the rules of precedence and associativity described in Question 1:

```
num zip num zip num zap num zap num zop num zop num
```

Question 3 (15 Points)

Student ID: ____________________

Consider the following grammar in Chomsky Normal Form:

$S \longrightarrow A\ B$
$S \longrightarrow B\ C$
$A \longrightarrow B\ A$
$A \longrightarrow a$
$B \longrightarrow C\ C$
$B \longrightarrow b$
$C \longrightarrow A\ B$
$C \longrightarrow a$

Show the complete CYK parsing table for the string `baaba` and indicate whether or not the string is in the language.

Question 4 (15 Points)

Student ID: ____________________

Write out the FIRST and FOLLOW sets for each nonterminal in the grammar below

$S \longrightarrow$ **void id lpar** M **rpar**
$M \longrightarrow P$
$M \longrightarrow \varepsilon$
$P \longrightarrow$ **id id comma** P
$P \longrightarrow$ **id id**

Question 5 (14 Points)

Student ID: ____________________

Consider the language Easy, recognized by the following grammar:

1) *Program* $\longrightarrow$ *Decls StmtList*
2) *Decls* $\longrightarrow$ *Type* **id** **;** *Decls*
3) *Decls* $\longrightarrow \varepsilon$
4) *Type* $\longrightarrow$ **int**
5) *Type* $\longrightarrow$ **double**
6) *StmtList* $\longrightarrow$ *Stmt* **;** *StmtList*
7) *StmtList* $\longrightarrow \varepsilon$
8) *Stmt* $\longrightarrow$ **id gets** *Exp*
9) *Exp* $\longrightarrow$ *Exp* **minus** *Exp*
10) *Exp* $\longrightarrow$ *Exp* **plus** *Exp*
11) *Exp* $\longrightarrow$ **id**
12) *Exp* $\longrightarrow$ **intlit**

For each production, write an SDT rule to accomplish the following SDT goal: The program should translate to the set of all **id**s that are accessed in a statement but have **not** been declared with a type.

Question 6 (14 Points)

Student ID: ____________________

Draw an AST and annotate each node with its type (or N/A if it has no type) for the unoptimized cRAP code below. You do not have to perfectly recall each of the AST node types, just draw a reasonable AST for the program.

```
int v;
int mathy(){
    int f;
    bool g;
    int h;
    f = 1 * 2 + 3;
    g = true and false;
    h = if (v > 4 - 2) { mathy(); }
    v = v - 1;
    return 2;
}
int main(){
    v = 2;
    mathy();
}
```

Question 7 (14 Points)

Student ID: ____________________

Draw a Control-Flow Graph for the unoptimized cRAP code below. Basic blocks should hold 3AC representing the code they contain.

```
void funk(){
  int mvar;
  >> mvar;
  if (mvar > 2){
    int kvar;
    mvar = mvar * 2;
    kvar = mvar;
  } else {
    mvar = mvar * 2;
  }
  << mvar;
}
```

Question 8 (14 Points)

Student ID: ____________________

Describe Static Single Assignment (SSA) form. What is/are the main advantages of translating IR code to SSA?

QUESTION 9 (15 POINTS)

Student ID: ____________________

To set up and break down an activation record, certain actions need to be taken in the caller (before and after the call) and in the callee at entry and exit. List out each of the steps that are needed for an arbitrary function (i.e. one that has arguments, locals, its own callees, a return value, etc). Write out the steps in order, and indicate where in the action should be taken.

Question 10 (15 Points)

Student ID: ____________________

Write out the MIPS code for the below function. You may use any of the optimizations that we discussed in class. Write out the names of any optimizations that you use.

```
void cool(){
  int a;
  int b;
  a = 8;
  b = 12;
  if (b > 2){
    int i;
    << b;
    i = 9;
  } else {
    >> b;
    a = b * 2;
  }
  << a;
}
```

Question 11 (15 Points)

Student ID: ____________________

A software company is offering two form of bug detectors. The first bug detector, Detector A, is complete but unsound. What style of analysis discussed in class might the company be using to implement this product (justify your answer)?

The same company is offering a second bug detector, Detector B, that is sound but incomplete. What type of analysis discussed in class might the company be using for to implement this product (justify your answer)?

The company claims that either detector could be integrated into an overnight compiler script that runs on a powerful build server. However, only one of these detectors is capable of acheiving this behavior based on the characteristics of the underlying analysis. Which bug detector could work this way? Why can't the other bug detector achieve the same results?

Question 12 (14 Points)

Student ID: ____________________

The 2nd Futamura projection uses a specializer to produce a compiler. However, the 1st Futamura projection uses the specializer as a compiler. What is the advantage of the 2nd projection given that the same net effect can be accomplished through the 1st projection?