CSE431 Final Examination
DUE: May 6th, 6:00PM

Name: _____

ID Number: _____

- Show your work to receive maximum credit. Partial credit will be given, but only if work is shown.
- Circle or otherwise indicate your answer to each question.
- Please keep your written answers brief and to the point. Incorrect or rambling statements can hurt your score on a question.
- If a question requires you to provide code, please use Java syntax. Minor syntax errors will not be counted against you, but extremely sloppy code will result in deductions.
- This exam is worth 20% of your final grade.
- The exam is broken up into 6 parts and is worth 100 points. There are 35 questions total. It is your responsibility to make sure you have all of the pages.
- You must sign the pledge below for your exam to count. Any cheating will cause the students involved to receive an F for this course. Other action may be taken.

Section	Possible Points	Score
I	15	
II	20	
III	15	
IV	15	
V	15	
VI	20	
Total	100	

Pledge: On my honor, I have neither given nor received any unauthorized aid on this exam.

Signed: _____

Preface: For this exam, you are tasked with adding exceptions to the compiler that we have been creating this semester. All of your answers must be submitted on this exam, however you may find it useful to revisit the labs and use them to help you answer the following questions.

1.) (15 points) Construct an AST for the following section of code:

```
try {
    if (v < 0){
        throw new ArithmeticException();
    } else {
        doSomething();
    }
} catch (Exception e) {
    return 0;
}
```

Make sure each node is labelled with a node type, and include all properties for each node. Pay special attention to any new node types that you have created to handle exceptions.

2.) (20 points) How would the grammar from lab 3 have to be modified to parse try/catch statements and throw statements? Show any rules that would need to be added or modified as well as the semantic actions necessary to create the AST for these rules.

3.) (15 points) Go through the type checking process for the AST you created in problem 1. List each type checking step in order, giving as much detail as possible.

4.) (15 points) Explain how semantic analysis would be applied to try/catch statements and throw statements. How do we know if a try statement terminates normally? If a catch statement terminates normally? Be sure to cover all possible cases.

5.) (15 points) Explain the structure of Exceptions within the JVM. Where do exceptions live? How are exceptions accessed? How are they created?

6.) (20 points) Create visit() methods that will generate JVM/Jasmin code for throwing exceptions and try/catch statements. Use the node types you created in problem 1.