# CS502: Compiler Design

Fall 2022 (Due: September 3rd, 2022)

Assignment A1: How you doing?

---

## 1 Assignment Objective

Use JavaCC and JTB to extend a programming language and traverse over parse trees.

## 2 Helping Joey!

Joseph Tribbiani (yes, the one from your favorite sitcom "FRIENDS") is working in an IT company now. As you all know he doesn't know his way around computers or its working. Joey is assigned in a compilers project and need your help to finish it. So help him finish his first task by completing the "JTLang". Also teach him how to count things by traversing parse trees.

## 3 Detailed Specification

You are provided with a grammar file `JT.jj`, which models a Java-like object-oriented programming language, consisting of classes, objects, integer variables and arrays, while loops, etc. Your first task is to expand the .jj file to include few more language features, as listed below.

- Types: A double precision floating-point type: `double`
- Algebraic operations: Addition (+) and Multiplication (`*`)
- Logical operations: AND (`&&`) and NEQ (`!=`)
- Object-oriented: inheritance (`extends`)
- Array-related: array-lookup (`Identifier[Identifier]`), array-length (`Identifier.length`)

Once you have completed the language `JTLang`, you need to generate a parser, and write a visitor to print the following: For each class, for each method, print (in **lexicographic order**):

- Classname : ParentClassname
- Classname.methodname
- No. of parameters.
- No. of local variables excluding parameters.
- No. of binary operations done.

### 3.1 Example Input

```
class Test {
    public static void main(String[] args) { //no of params = 1
        System.out.println(new A().foo(10));
    }
}
class A {
    public int foo(int p) { //no of params = 1
        int x; //local var1
        double y; //local var2
        boolean b; //local var3
        int r; //local var4
        x = 10;
        y = 100.25;
        b = true;
        while (b) {
            y = y + x; //binop1
            if (y != 200) { //binop2
                b = false;
            }
        }
        r = this.bar();
        return r;
    }
    public int bar() { //no of params = 0
        int a; //local var1
        int b; //local var2
        int c; //local var3
        int d; //local var4
        a = 10;
        b = 20;
        c = a + b; //binop1
        d = a - b; //binop2
        if(d != (a * c)){ //binop3 //binop4
            a = c * d; //binop5
        }
        return a;
    }
}
class B extends A {}
```

### 3.2 Example Output

```
Class A :
Method A.bar 0 4 5
Method A.foo 1 4 2
Class B : A
Class Test :
Method Test.main 1 0 0
```

### 3.3 Evaluation

Your submission must be named as `rollnum-a1.zip`, where `rollnum` is your roll-number in small letters. Upon unzipping the submission, we should get a directory named `rollnum-a1`. The main class inside this directory should be named `Main.java`. Your program should read from the standard input and print to the standard output. You can leave all the visitors and syntax-tree nodes as it is, but remember to remove all the `.class` files and jar files.

We would run the following commands as part of the automated evaluation process:

- `javac Main.java`
- `java Main < test > out`

If the contents of `out` match with the expected output for the testcase, you would get marks for the corresponding testcase.

## 4 Plagiarism Warning

You are allowed to discuss publicly on class, but are supposed to do the assignment completely individually. We would be using sophisticated plagiarism checkers, and if similarity is found, the penalty used in the course would be as follows:

- First instance: 0 marks in the assignment
- Second instance: Grade reduction.
- Third instance: F grade and report to disciplinary committee.

-*-*-*- Do the assignment honestly; enjoy learning the course. -*-*-*-