```
This is coursework 1 model answers.

CW1.1

This is one solution:

TOKEN :
{
  < INTEGER_LITERAL:
        <DECIMAL_LITERAL> ("L")?
      | <HEX_LITERAL> ("L")?
      | <OCTAL_LITERAL> ("L")?
  >
|
  < #DECIMAL_LITERAL: ("0" | (["1"-"9"] (["0"-"9"])*)) >
|
  < #HEX_LITERAL: "0" ["x","X"] (["0"-"9","a"-"f","A"-"F"])+ >
|
  < #OCTAL_LITERAL: "0" (["0"-"7"])+ >
|
  < FLOATING_POINT_LITERAL:
        <DIGITS> "." (<DIGITS>)? (<EXPONENT>)? (<TYPESUFFIX>)?
      | "." <DIGITS> (<EXPONENT>)? (<TYPESUFFIX>)?
      | <DIGITS> <EXPONENT> (<TYPESUFFIX>)?
      | <DIGITS> (<EXPONENT>)? <TYPESUFFIX>
  >
|
  < #DIGITS: (["0"-"9"])+ >
|
  < #TYPESUFFIX: (["f","d"]) >
|
  < #EXPONENT: "e" (["+","-"])? (["0"-"9"])+ >
}


CW1.2

The new/changed files are: minijava.jj,
syntaxtree/{DoWhile,Try,Catch,Throw,CatchList}.java,
visitor/Visitor.java, visitor/DBPrettyPrint.java

----------------minijava.jj-------------------------
Token definitions must be added:

| < DO: "do" >                  /* Added by DB */
| < TRY: "try" >                /* Added by DB */
| < FINALLY: "finally" >        /* Added by DB */
| < THROW: "throw" >            /* Added by DB */
| < CATCH: "catch" >            /* Added by DB */

Then:

Statement Statement():
......
|                               /* Added by DB */
  s=DoWhileStatement()          /* Added by DB */
|                               /* Added by DB */
  s=TryStatement()              /* Added by DB */
|                               /* Added by DB */
  s=ThrowStatement()            /* Added by DB */
....
}
...

Statement DoWhileStatement() :
{
  Statement s;
  Exp e;
}
{
  "do" s=Statement() "while" "(" e=Expression() ")" ";"
  { return new DoWhile(s,e); }
}

Statement TryStatement() :
{
  Statement s;
  StatementList sl1 = new StatementList(), sl2 = new StatementList();
  Catch c;
  CatchList cl = new CatchList();
```

```
}
{
  "try" "{" ( s=Statement() { sl1.addElement(s); } )* "}"
  ( c=Catch() { cl.addElement(c); } )*
  "finally"  "{" ( s=Statement() { sl2.addElement(s); } )* "}"
  { return new Try(sl1,cl,sl2); }
}

Statement ThrowStatement() :
{
  Exp e;
}
{
  "throw" e=Expression() ";"
  { return new Throw(e); }
}

Catch Catch() :
{
  Type t;
  Identifier i;
  Statement s;
  StatementList sl=new StatementList();
}
{
  "catch" "(" t=Type() i=Identifier() ")"
    "{" ( s=Statement() { sl.addElement(s); } )* "}"
  { return new Catch(t,i,sl); }
}

-----------------end of minijava.jj additions----------------------

-----------------DoWhile.java-------------------------------------

public class DoWhile extends Statement {        /* Added by DB */
  public Statement s;
  public Exp e;

  public DoWhile(Statement as, Exp ae) {
    s=as; e=ae;
  }

  public void accept(Visitor v) {
    v.visit(this);
  }
}
----------------end of DoWhile.java-----------------------------
----------------Try.java----------------------------------------
public class Try extends Statement {    /* Added by DB */
  public StatementList sl1;
  public CatchList  cl;
  public StatementList sl2;

  public Try(StatementList ts, CatchList cs, StatementList fs) {
    sl1=ts; cl=cs; sl2=fs;
  }

  public void accept(Visitor v) {
    v.visit(this);
  }
}
-----------------end of Try.java---------------------------------
-----------------Throw.java--------------------------------------
public class Throw extends Statement {  /* Added by DB */
  public Exp e;

  public Throw(Exp te) {
    e=te;
  }

  public void accept(Visitor v) {
    v.visit(this);
  }
}
------------------end of Throw.java-----------------------------
------------------Catch.java------------------------------------
public class Catch {    /* Added by DB */
  public Type t;
  public Identifier i;
  public StatementList sl;
```

```
  public Catch(Type ct, Identifier ci, StatementList cs) {
    t=ct; i=ci; sl=cs;
  }

  public void accept(Visitor v) {
    v.visit(this);
  }
}
-------------------end of Catch.java----------------------------
-------------------CatchList.java-------------------------------
public class CatchList {        /* Added by DB */
   private Vector list;

   public CatchList() {
      list = new Vector();
   }

   public void addElement(Catch n) {
      list.addElement(n);
   }

   public Catch elementAt(int i)  {
      return (Catch)list.elementAt(i);
   }

   public int size() {
      return list.size();
   }
}
---------------------end of CatchList.java------------------------

In visitor/Visitor.java, added to interface Visitor:

  public void visit(DoWhile n);         /* Added by DB */
  public void visit(Try n);             /* Added by DB */
  public void visit(Throw n);           /* Added by DB */
  public void visit(Catch n);           /* Added by DB */

In visitor/DBPrettyVisitor.java (the do-while is not particularly
pretty, students should get good marks for doing better
than this):

  // Statement s;
  // Exp e;
  public void visit(DoWhile n) {
    System.out.print(indent+"do");
    if (!(n.s instanceof Block)) {
      System.out.println();
      indent=indent+"  ";
    }
    n.s.accept(this);
    if (!(n.s instanceof Block))
      indent=indent.substring(2);
    System.out.println();
    System.out.print(indent+"while ( ");
    n.e.accept(this);
    System.out.print(" );");
  }

  // StatementList sl1;
  // CatchList cl;
  // StatementList sl2;
  public void visit(Try n) {
    System.out.println(indent+"try {");
    indent=indent+"  ";
    for ( int i = 0; i < n.sl1.size(); i++ ) {
        n.sl1.elementAt(i).accept(this);
        System.out.println();
    }
    indent=indent.substring(2);
    System.out.println(indent+"}");
    if (n.cl.size() > 0) {
      for ( int i = 0; i < n.cl.size(); i++ ) {
          n.cl.elementAt(i).accept(this);
          System.out.println();
      }
    }
    System.out.println(indent+"finally {");
    indent=indent+"  ";
```

```
    for ( int i = 0; i < n.sl2.size(); i++ ) {
        n.sl2.elementAt(i).accept(this);
        System.out.println();
    }
    indent=indent.substring(2);
    System.out.print(indent+"}");
  }

  // Type t;
  // Identifier i;
  // StatementList e;
  public void visit (Catch n) {
    System.out.print(indent+"catch ( ");
    n.t.accept(this);
    System.out.print(" ");
    n.i.accept(this);
    System.out.print(" ) {");
    System.out.println();
    indent=indent+"  ";
    for ( int i = 0; i < n.sl.size(); i++ ) {
        n.sl.elementAt(i).accept(this);
        System.out.println();
    }
    indent=indent.substring(2);
    System.out.print(indent+"}");
  }

  // Exp e;
  public void visit (Throw n) {
    System.out.print (indent+"throw ");
    n.e.accept(this);
    System.out.print (";");
  }

End of model answers
```