**public** **static** Term fromString(String str){

String temp = **new** String(str);

TermImp result = **null**;

**if** (temp.contains("x^")){

// handle term with the form ax^n

StringTokenizer strTok = **new** StringTokenizer(temp, "x^");

List<String> list = **new** ArrayList<String>(2);

**while**(strTok.hasMoreElements()){

list.add((String) strTok.nextElement());

}

**if** (list.size() == 0){

**throw** **new** IllegalArgumentException("Argument string is formatter illegally.");

}

**else** **if** (list.size() == 1){

// term if of the form x^n, where n is the exponent

Integer expo = Integer.*parseInt*(list.get(0));

result = **new** TermImp(1, expo);

}

**else** {

// term if of the form ax^n, where a, (a != 1) is the coefficient and n is the exponent

Double coeff = Double.*parseDouble*(list.get(0));

Integer expo = Integer.*parseInt*(list.get(1));

result = **new** TermImp(coeff, expo);

}

}

**else** **if** (temp.contains("x")){

// handle value with exponent == 1

StringTokenizer strTok = **new** StringTokenizer(temp, "x");

List<String> list = **new** ArrayList<String>(2);

**while**(strTok.hasMoreElements()){

list.add((String) strTok.nextElement());

}

**if** (list.size() == 0){

// term is of the form x, with coefficient = 1 and exponent = 1

result = **new** TermImp(1.0, 1);

}

**else** {

// term is of the form ax, with coefficient = a and exponent = 1

Double coeff = Double.*parseDouble*(list.get(0));

result = **new** TermImp(coeff, 1);

}

}

**else** {

// handle numeric value

result = **new** TermImp(Double.*parseDouble*(temp), 0);

}

**return** result;

}