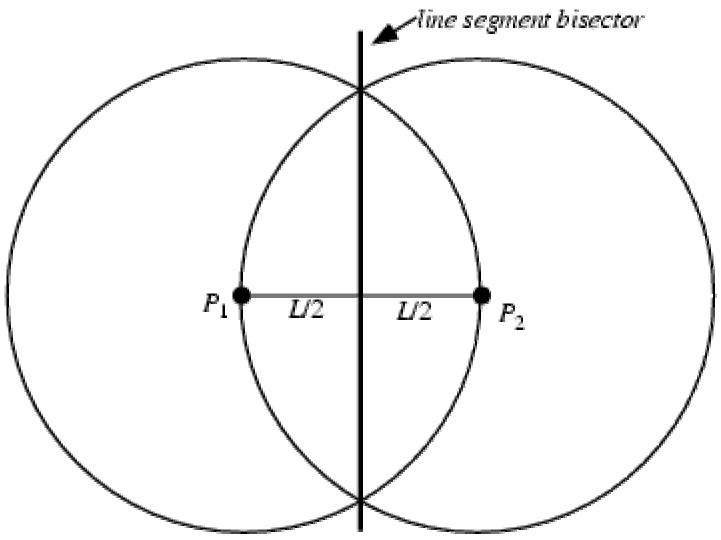
School Geometry Description Language – G++

Columbia University
COMS W4115 Programming Languages and Translators
Fall 2006

Prof. Stephen Edwards
Author:
Sanjay Kumar
sanjayk98@gmail.com
15-Dec-2006



Abstract

The most important challenge of e-learning software for school math is to develop a computer program, which can act as a teacher. It can generate question and test the skill set of students. Such automated testing will require knowledge-domain specific computer language to describe the concept and solve it. We have attempted to develop one such language for school geometry. We will call this G++

1. Introduction

The idea of language to aid creating and visualizing geometric object is not new. AutoCAD is one of successful product in this area. However, it is more suited for civil engineer and architect than a high school student learning geometry. We need a computer language, which can help student to prove theorem algorithmically and solve geometric puzzle.

The goal of describing geometric concept using natural language is very attractive but it is very hard to achieve with limited time and resource provided for course project. Therefore, it was necessary to simplify the scope of project. We made following simplification for the purpose of this project. We will follow strict syntax to describe the geometry question and its concept. The deliverable of this project will not include any of geometric construction and graphics. Conditional statement and loop around the statement will also be excluded from the project. We will not attempt to derive new knowledge algorithmically but use simple formula to compute relationship or derived attribute. Despite above restrictions, language can be very useful to describe a large majority of geometric objects, their attributes and relationships. The input of G++ compiler will be text for question and output will be a nicely formatted PDF document with question and answer.
Sample Applications: Following are some of sample applications. Please, note that not all functionalities are implemented in the current implementation.

Computing derived value: The most of geometry questions are asked to derive value based on fomula or some theorem. It is possible to compute these value using compiler build-in support for basic types.

***Generation of model answer*:** It is often required by teachers to publish a model solution for questions assigned to students in class. G++ compiler will be capable of generating nicely formatted solution for various geometry questions.

*Visual display of geometric shape***:** It is often hard for teachers to draw various complex geometric shapes as a document. The popular authoring software like MS word etc. is not equipped to deal with the requirement of a Geometry teacher. G++ compiler will be able to construct geometric shape with ease. Although, it is beyond the scope of the current project, drawing of few basic shapes are demonstrated in this report.

Movie play of theorem proving: Although, it is not in scope of current project, it may be relatively easy to add support for flash file generation, which will show movie for better learning experience.

2. Language Tutorial

Below are typical geometry question and required code to describe the same in G++.

Q1: Calculate the area of circle with radius 78 cm.

Translation into G++
Given circle: a; radius of a=78; find area of a;

Q2: Find area of triangle with side 10, 15 and 12 cm.

Translation into G++
Given line: ab, bc, ca, Triangle: abc; length of ab=10, length of bc=15, length of ca=12 ; find area of abc;

As it is evident from above code, code in G++ is almost similar to natural language question and it can be very easily learned by novice computer users. The simplicity and tolerance to users error is very important part of G++ as it is meant for school children and teachers, which may have very little experience of computing.

2.1 Learning G++ language basics

Any geometry question has basically three parts.

a. Declare section: Given objects: Object can be described in G++ by simply specifying the type and name. Below is example:

Example 1: Given circle a;
Above statement will define a circle named a; G++ is case insensitive and therefore, one can mix and match different case for better readability.

Example 2: Given Circle A, B, Point C;
One can define multiple objects separated by comma. Declare section ends with semi-comma; Language supports many basic geometric shapes like circle, point etc and can be extended further by adding new library file (Section …)

b. Fact Section: Facts about objects: Usually, we define different fact about the object in geometry. Fact like radius of circle is 5 cm etc. This can be easily described using G++.

Example 1: radius of a=5;
The language will automatically determine the object type for named variable a and find out if radius is a compatible attribute or not. Also, one can assign two-dimensional value using one expression.

Example 2: location of c=(8,10);
Multiple facts can be stated in one expression if they are separated by comma.

Example 3: location of c=(8,20), radius of a=5;

c. Question section: Question about relationship of objects or derivation of attribute value Relationship or derived attribute value can be asked in third section of question.

Example 1: Find area of a;
As it is evident from above example, this section starts with keyword 'find'. It is followed by derived attribute or relationship name. One can ask multiple questions separated by comma.

Example 2: Find area of a, perimeter of a;
Optionally, one can ask to draw the object like 'draw a';
Putting everything together below is a complete example:

Complete example:

Example 1. Find the area, perimeter of a circle with radius of 5cm.
G++ code
Given circle: a; radius of a=5; find area of a, perimeter of a;

Example 2. Find the distance between two point (10,10) and (70,10)
G++ code
Given point: a, b,line: ab; location of a=(10,10), location of b=(70,10); find length of ab;

2.2 Learning advanced feature of G++

Although, G++ provides library file for all major geometric shapes, it is possible to extend it further by adding new type or adding new attributes or relationship of objects. This extensibility is meant for experienced user, who wants to extend the language to meet their need. The concept described in this section can be easily understood in examples. Below is example for defining Point.

Shape: point;
attribute: x, type number, default 100;
attribute: y, type number, default 100;
derived attribute: definition, type text, default "a geometric element that has position but no extension; a point is defined by its coordinates" ;
derived attribute: angle, type number, default expr("Math.atan(x/y)");
global relation: find_distance refers point a, point b returns number d   formula d=expr("Math.sqrt((a.x-b.x)*(a.x-b.x)+ (a.y-b.y)*(a.y-b.y))");
;

Above definition has three section. Name of object_type in shape section, attribute/derived attribute and relation/global relation. Below is description of each section.

Shape: Here, we name the basic shape type. Example of shape is circle, point, line etc. It could be any name with literal value.

Attribute: Attribute is feature of shape, which defines the object. The name of attribute can be any alphabetic character followed by type and default value. There are three basic types possible for attribute. *Text, number or instance of shape*

Default: Default value could be a number or pair of number or literal or a java expression. Example: default 100, default (20,20), default (x,y), default expr("2*pi*r) etc.

Derived attribute: The format of derived attribute is same as attribute. While attribute is mandatory to define an object, derived attribute is optional.
Relation: Relation is association among two or more number of objects. It computes the value for relation using some set of formulas and returns number, text or object.

Global Relation: Global relation is similar to relation except it is implemented as static method in java.

Below is example for defining Circle.

Shape: circle;
attribute: radius, type number, default 100;
attribute: center, type point, default (0,0);
derived attribute: definition, type text, default "In Euclidean geometry, a circle is the set of all points in a plane at a fixed distance, called the radius, from a fixed point, called the centre. Circles are simple closed curves, dividing the plane into an interior and exterior. Sometimes the word circle is used to mean the interior, with the circle itself called the circumference. Usually however, the circumference means the length of the circle, and the interior of the circle is called a disk or disc. ";
derived attribute: perimeter, type number, default expr("2*Math.PI*radius");
derived attribute: area, type number, default expr("Math.PI*radius*radius");
derived attribute: diameter, type number, default expr("2*radius");
global relation: find_distance refers circle a, circle b returns number distance
formula distance=expr("Point.find_distance(a.center,b.center)") ;

Expr: Expr is special keyword in the language and it can be used to assign any java expression to G++ variable. G++ does not validate syntax within expression and therefore, one would get java compilation error if its syntax is incorrect.

Also, you can notice that we have made use of global relation find_distance, which was defined in previous library file for Point. This provides modularity for the language to compute attribute of a shape, which is dependent on many other objects. “Refers” keyword is followed by list of objects passed to this function and “returns” keyword is followed by data type returned by this function. These are simple way of describing input and output for java function, which will be generated as intermediate language to meet the computing need of G++. It is also possible to assign default value for derived attribute as java expression.

3. Language Reference Manual

Introduction:

G++ language has two different syntax
1. To describe school geometry question.
2. To enrich the language by adding support for additional geometric shape or attribute. We shall herein refer first part as G++ question and second part as G++ library.

3.1 Lexical conventions

3.1.1 Token: There are five kinds of tokens: identifiers, keywords, constants, text, expression and other separators. In general blanks, tabs, new lines, comma, semi-comma, colon and comments as described below are ignored except as they serve to separate tokens. At least one of these characters is required to separate otherwise adjacent identifiers, constants and key word. If the input stream has been parsed into tokens up to a given character, the next token is taken to include the longest string of characters, which could possibly constitute a token.

3.1.2 Comments

The characters /* introduce a comment, which terminates with the characters */.

3.1.3 Identifiers (Names)

An identifier is a sequence of letters and digits; the first character must be alphabetic. The underscore "_" counts as alphabetic. *Upper and lower case letters are considered same.* There is no restriction on the length of identifier.

3.1.4 Keywords

The following identifiers are reserved for use as keywords, and may not be used otherwise:

Given	Find	Attribute
Derived attribute	Relation	Global relation
Shape	Of	Draw

3.1.5 Constants

pi

3.1.6 text: A string is a sequence of characters surrounded by double quotes " " ".

4. Syntax summary

```
<Question>: <declare_section> <fact_section> <question_section>
<declare_section>: “Given” (<shape> var_list)+
<fact_section>: (<attribute_name> of <var_name> = value)*
<question_section>: “find” (<derived_attribute> of <var_name>)+

<shape_definition>: <shape_name> ID
                    <attribute_name> ID, type <object_type> default   <value>
                    <derived_attribute_name>  ID, default <value>
                     <relation_name> ID refers  (<shape> ID)* returns ID
                      formula: (lvalue=rvalue)*
```

Above is summary of syntax. However, more detailed for grammar can be found in antlr grammar attached with this report. The best way to learn this language is through examples. Below are three examples, which use specified syntax.

Example 1: Define a geometric shape square

Square has one mandatory attribute called ‘side’ and could have any number of derived attribute like area, perimeter etc. If we need to find maximum radius of a circle inscribed into it then it can be defined a relation between current instance of square and a circle. Below is code for library file.

Shape: square;
attribute: side, type number, default 100;
derived attribute: definition, type text, default "a plane rectangle with four equal sides and four right angles; a four-sided regular polygon" ;
derived attribute: area, type number, default expr("side*side");
derived attribute: location, type point, default (0,0);
derived attribute: center, type point, default (50,50);
relation: setLocation refers number h_x
formula location.x=h_x;
relation: setLocationY refers number h_y
formula location.y=h_y;
relation: find_radius returns number r
formula r=side/2;

Here is generated java code with inline comment.

```
// standard library for geometry and io are included in header.
import java.awt.geom.*;
import java.awt.image.BufferedImage;
import java.awt.*;
import java.io.File;
import java.io.IOException;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import javax.imageio.ImageIO;
public class Square{
// this attribute is added by default. Value is changed at runtime
public String name="a";
// two constructors are generated by default. One with null argument
// and other with argument for mandatory attributes
Square(){}
Square(double h_side){
side=h_side;
}
//Attribute is always initialized with default value. Specifying the
//default value is mandatory. What will happen if default value is
//another instance of shape? See next example
public double side=100;
// set method is automatically generated for all mandatory and derived //
attribute
public  void setSide(double h_side){
side =h_side;
}
// definition text is optional and can be added for each shape.
public String definition="a plane rectangle with four equal sides and four right
angles; a four-sided regular polygon";
// For each derived attribute, one function is generated in java, which
// is specically designed for generating nicely formatted string as
// solution text for question.
public  String derive_Definition(){
String txt,s1,s2,s3;
s1="Definition="+"a plane rectangle with four equal sides and four right angles; a
four-sided regular polygon";
s2="Definition of "+this.getClass().getName().toLowerCase()+" "+name+"="+"a
plane rectangle with four equal sides and four right angles; a four-sided regular
polygon";
txt=s1+"\n"+s2;
return s1;
}
// java expression is decoded into actual java code
public double area=side*side;
// another auto-generated java function to generate derivation text for //area.
As it is evident from this example that number is automatically //formatted to
default format.
public  String derive_Area(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Area=side*side";
s2="Area of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(side*side);
txt=s1+"\n"+s2;
return txt;
}
// here default value is instance of another shape. Therefore, antlr
//code recognize this and assigns its value using default constructor
//for point.

public Point location=new Point(0,0);
//derivation text in current version does not know how to describe //instance
of shape type and therefore, it emits only partial solution //text. However, it is
possible to implement description of shape //instance in future release.
public  String derive_Location(){
String txt,s1,s2,s3;
s1="Location=new Point(0,0)";
s2="Location of "+this.getClass().getName().toLowerCase()+"
"+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
// another derivation. default (50,50) is automatically appended to
//constructor. Anlr implementation requires multiple pass for parsing. //See
antlr code for details.
public Point center=new Point(50,50);
public  String derive_Center(){
String txt,s1,s2,s3;
s1="Center=new Point(50,50)";
s2="Center of "+this.getClass().getName().toLowerCase()+" "+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
```

```
// location x is set. Although, (x,y) is assigned together in actual
//g++ code, we generate two functions
public void  setLocation(double h_x){

location.x=h_x;
}
//location y is set.
public void  setLocationY(double h_y){

location.y=h_y;
}
// relation is converted into java function. Variable and function
// argument is automatically generated. In this case, relation does not
// refer to any other object and therefore, argument is null.
public double  find_radius(){
double r ;
r=side/2;
return r;
}
}
```

Above code was example of generated code by library g++. Below is example of generated code for question g++. Question g++ makes use for shape and attribute defined by g++ library and make it available for end-user. Below is example of question g++ generated code.

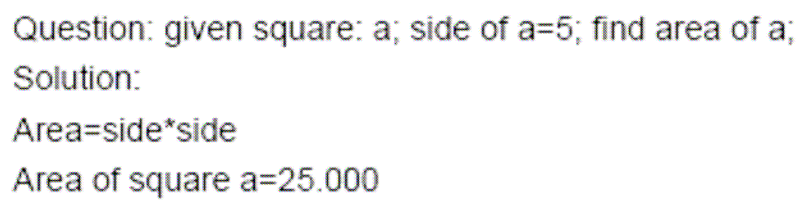
Question: Find the area of a square with side equal to 5 cm. Also, find maximum radius of circle inscribed into it.

G++ code

```
Given square: a; side of a=5; find area of a;
```

```
//Generated java code
//header file
import javax.swing.*;
import java.awt.*;
//Class name SGMLCircle is historical as it was first coded with this //name.
However, it works equally well for all kind of shapes.
public class SGMLCircle extends JPanel {
// support for drawing geometric shape was added in first version but //
removed later to limit the scope of this project.
 protected void clear(Graphics g) {
super.paintComponent(g);
}
public static void main(String[] args) {
//SGMLDocument is wrapper class for iText which supports the generation //
of pdf document.
SGMLDocument document=new SGMLDocument();
//Question is first added to pdf document.
document.add_solution_text("Question: given square: a; side of a=5;
find area of a;");
// Solution part begins here
document.add_solution_text("Solution:");
//All instance of geometic shape is initialized.
Square          a=new Square();
// name is automatically assigned which is same as variable name.
// Variable name is needed during generation of solution text.
a.name="a";
// facts are parsed and value is assigned by calling proper set method.
a.setSide(5);
// Finally, solution text is derived and added to pdf document.
document.add_solution_text(a.derive_Area());
document.close();
}
}
```

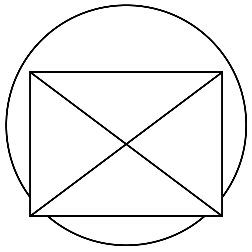
This is example of actual, pdf document generated by running above java



```
Question: given square: a; side of a=5; find area of a;
Solution:
Area=side*side
Area of square a=25.000
```

code.

Example 2: Define a square inscribed within a circle.



We shall call this shape as “sqaure_within_circle”

Below is definition
// Assuming that we have already defined square and circle, we can //define a composite shape called square_within_circle.
Shape: square_within_circle;
// The main attribute is definition of circle, rest attribute can be
// derived from this. Please, note that circle is initialized with
//default constructor with value 100. However, its value can be changed
//at runtime by user.

attribute: h_circle, type circle, default (100.0);
// Given the circle, we can derive the maximum size of square by //following formula
derived attribute: h_square, type square, default expr("new Square(h_circle.diameter/1.41)");
// other assosiated property of square can be easily exposed by just //naming it.
derived attribute: area_of_square, type number, default expr("h_square.area");
// deriving the area of circle
derived attribute: area_of_circle, type number, default expr("h_circle.area");
// We can do more complex calulation using formula and referring to //mutiple objects.
relation: find_side returns number h_side
    formula h_side=expr("h_square.side");
;

Below is generated code.

```
//header
import java.awt.geom.*;
import java.awt.image.BufferedImage;
import java.awt.*;
import java.io.File;
import java.io.IOException;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import javax.imageio.ImageIO;
//name of class is automatically generated. It should be added to //permanent
symbol file.
 public class Square_within_circle{
public String name="a";
Square_within_circle(){}
Square_within_circle(Circle h_h_circle){
h_circle=h_h_circle;
}
public Circle h_circle=new Circle(100.0);
//we can assign any circle
public  void setH_circle(Circle h_h_circle){
h_circle =h_h_circle;
}
public Square h_square=new Square(h_circle.diameter/1.41);

public  String derive_H_square(){
String txt,s1,s2,s3;
s1="H_square=new Square(h_circle.diameter/1.41)";
s2="H_square of "+this.getClass().getName().toLowerCase()+"
"+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
public double area_of_square=h_square.area;
//derivation for area of square
public  String derive_Area_of_square(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Area_of_square=h_square.area";
s2="Area_of_square of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(h_square.area);
txt=s1+"\n"+s2;
return txt;
}
//derivation for area of circle
public double area_of_circle=h_circle.area;
public  String derive_Area_of_circle(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Area_of_circle=h_circle.area";
s2="Area_of_circle of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(h_circle.area);
txt=s1+"\n"+s2;
return txt;
}

//more complex calulcation can be done here
public double  find_side(){
double h_side ;
h_side=h_square.side;
return h_side;
}
}
```

Assuming that above library file is added to g++, g++ question will be ready to exploit newly defined shape and its attribute. Below is a sample question and its crossponding code in g++
Q. given circle of radius 67 cm, find the area of square inscribed within it.

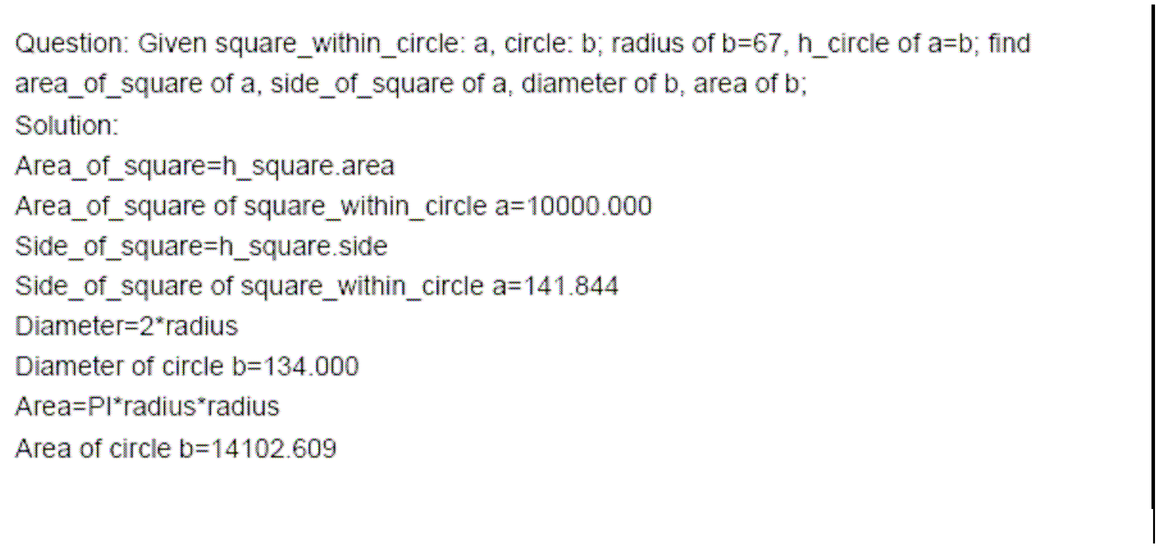
G++ code

```
//here is few lines of g++ code
SGML> Given square_within_circle: a, circle: b; radius of b=67, h_circle of
a=b;find area_of_square of a, side_of_square of a, diameter of b, area of b;
```

Generated java code

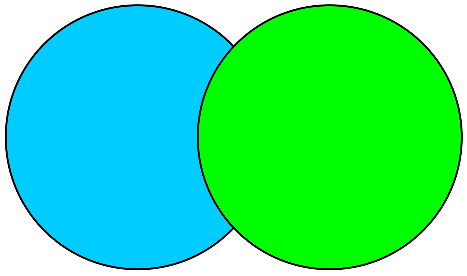
```
// few lines of g++ code tranlates into hundred lines of java code
import javax.swing.*;
import java.awt.*;
public class SGMLCircle extends JPanel {
 protected void clear(Graphics g) {
super.paintComponent(g);
}
public static void main(String[] args) {
SGMLDocument document=new SGMLDocument();
//question is added to first line of pdf document
document.add_solution_text("Question: Given square_within_circle: a, circle: b;
radius of b=67, h_circle of a=b;");
document.add_solution_text("Solution:");
//here, we start the solution
Square_within_circle          a=new Square_within_circle();
a.name="a";
//define circle and set radius to 67
Circle   b=new Circle();
b.name="b";
b.setRadius(67);
// assign new geometric shape property to circle
a.setH_circle(b);
// add derivation text for various question to pdf
document.add_solution_text(a.derive_Area_of_square());
document.add_solution_text(a.derive_Side_of_square());
document.add_solution_text(b.derive_Diameter());
document.add_solution_text(b.derive_Area());
document.close();
}
}
```

Below is image of pdf document generated by java code. There is some nasty bug still left in the code, which generates incorrect value for nested value assignment.



```
Question: Given square_within_circle: a, circle: b; radius of b=67, h_circle of a=b; find
area_of_square of a, side_of_square of a, diameter of b, area of b;
Solution:
Area_of_square=h_square.area
Area_of_square of square_within_circle a=10000.000
Side_of_square=h_square.side
Side_of_square of square_within_circle a=141.844
Diameter=2*radius
Diameter of circle b=134.000
Area=PI*radius*radius
Area of circle b=14102.609
```

Example 3: Find the area of intersection between two circle each having radius of 45 cm and seperated by 60 cm.



Definition of two circle

```
Shape: two_circle;
Attribute: circle1, type circle, default (45);
Attribute: circle2, type circle, default (45);
Attribute: center1, type point, default (0,0);
Attribute: center2, type point, default (60,0);
Derived attribute: distance, type number, default  expr("center2.x-center1.x");
Derived attribute: p1, type point, default expr("intersection_p1()")  ;
Derived attribute: p2, type point, default expr("intersection_p2()")  ;
Derived attribute: c1, type point, default expr("circle1.center");
Derived attribute: c2, type point, default expr("circle2.center");
Derived attribute: triangle1, type triangle, default (c1,p1,p2) ;
Derived attribute: triangle2, type triangle, default (c2,p1,p2) ;
Derived attribute: triangle1_area, type number, default expr("triangle1.area") ;
Derived attribute: triangle2_area, type number, default expr("triangle2.area")  ;
Derived attribute: arc1, type number, default
expr("2*Math.asin(Point.find_distance(p1,p2)/2*circle1.radius)");
Derived attribute: arc2, type number, default
expr("2*Math.asin(Point.find_distance(p1,p2)/2*circle2.radius)");
Derived attribute: arc1area, type number, default
expr("arc1*circle1.area/2*Math.PI");
Derived attribute: arc2area, type number, default
expr("arc2*circle2.area/2*Math.PI");
Derived attribute: intersection_area, type number, default
expr("arc1area+arc2area-triangle1.area-triangle2.area");
Relation:  intersection_p1 returns point p11
```

```
          Formula
p11.x=expr("center1.x+(distance*distance+circle1.radius*circle1.radius-
circle2.radius*circle2.radius)/2*distance");
                p11.y= expr("center1.y+Math.sqrt(circle1.radius*circle1.radius-
(p11.x-center1.x)*(p11.x-center1.x))");
Relation:  intersection_p2 returns point p22
          Formula
p22.x=expr("center2.x+(distance*distance+circle2.radius*circle2.radius-
circle1.radius*circle1.radius)/2*distance");
                p22.y= expr("center2.y+Math.sqrt(circle2.radius*circle2.radius-
(p22.x-center2.x)*(p22.x-center2.x))");
Relation: find_area returns number a
        formula a=intersection_area;
Relation: find_arc1area returns number a
        formula a=arc1area;
Relation: find_arc2area returns number a
        formula a=arc2area;
Relation: find_arc1 returns number a
        formula a=arc1;
Relation: find_arc2 returns number a
        formula a=arc2;
Relation: intesection_p1 returns point a
        formula a=expr("intersection_p1()");
Relation: intesection_p2 returns point b
        formula b=expr("intersection_p2()");
        ;
```

Generated intermediate java code

```
import java.awt.geom.*;
import java.awt.image.BufferedImage;
import java.awt.*;
import java.io.File;
import java.io.IOException;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import javax.imageio.ImageIO;

 public class Two_circle{
public String name="a";
Two_circle(){}
public Circle circle1=new Circle(45);
public  void setCircle1(Circle h_circle1){
```

```
circle1 =h_circle1;
}
public Circle circle2=new Circle(45);
public  void setCircle2(Circle h_circle2){
circle2 =h_circle2;
}
public Point center1=new Point(0,0);
public  void setCenter1(Point h_center1){
center1 =h_center1;
}
public Point center2=new Point(60,0);
public  void setCenter2(Point h_center2){
center2 =h_center2;
}
public double distance=center2.x-center1.x;
public  String derive_Distance(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Distance=center2.x-center1.x";
s2="Distance of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(center2.x-center1.x);
txt=s1+"\n"+s2;
return txt;
}
public Point p1=intersection_p1();
public  String derive_P1(){
String txt,s1,s2,s3;
s1="P1=intersection_p1()";
s2="P1 of "+this.getClass().getName().toLowerCase()+" "+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
public Point p2=intersection_p2();
public  String derive_P2(){
String txt,s1,s2,s3;
s1="P2=intersection_p2()";
s2="P2 of "+this.getClass().getName().toLowerCase()+" "+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
public Point c1=circle1.center;
public  String derive_C1(){
```

```
String txt,s1,s2,s3;
s1="C1=circle1.center";
s2="C1 of "+this.getClass().getName().toLowerCase()+" "+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
public Point c2=circle2.center;
public  String derive_C2(){
String txt,s1,s2,s3;
s1="C2=circle2.center";
s2="C2 of "+this.getClass().getName().toLowerCase()+" "+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
public Triangle triangle1=new Triangle(c1,p1,p2);
public  String derive_Triangle1(){
String txt,s1,s2,s3;
s1="Triangle1=new Triangle(c1,p1,p2)";
s2="Triangle1 of "+this.getClass().getName().toLowerCase()+"
"+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
public Triangle triangle2=new Triangle(c2,p1,p2);
public  String derive_Triangle2(){
String txt,s1,s2,s3;
s1="Triangle2=new Triangle(c2,p1,p2)";
s2="Triangle2 of "+this.getClass().getName().toLowerCase()+"
"+name+"="+"todo";
txt=s1+"\n"+s2;
return txt;
}
public double triangle1_area=triangle1.area;
public  String derive_Triangle1_area(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Triangle1_area=triangle1.area";
s2="Triangle1_area of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(triangle1.area);
txt=s1+"\n"+s2;
return txt;
}
```

```
public double triangle2_area=triangle2.area;
public  String derive_Triangle2_area(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Triangle2_area=triangle2.area";
s2="Triangle2_area of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(triangle2.area);
txt=s1+"\n"+s2;
return txt;
}
public double arc1=2*Math.asin(Point.find_distance(p1,p2)/2*circle1.radius);
public  String derive_Arc1(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Arc1=2*asin(Point.find_distance(p1,p2)/2*circle1.radius)";
s2="Arc1 of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(2*Math.asin(Point.find_distance(p1,p2)/2*circle1.
radius));
txt=s1+"\n"+s2;
return txt;
}
public double arc2=2*Math.asin(Point.find_distance(p1,p2)/2*circle2.radius);
public  String derive_Arc2(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Arc2=2*asin(Point.find_distance(p1,p2)/2*circle2.radius)";
s2="Arc2 of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(2*Math.asin(Point.find_distance(p1,p2)/2*circle2.
radius));
txt=s1+"\n"+s2;
return txt;
}
public double arc1area=arc1*circle1.area/2*Math.PI;
public  String derive_Arc1area(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Arc1area=arc1*circle1.area/2*PI";
s2="Arc1area of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(arc1*circle1.area/2*Math.PI);
```

```
txt=s1+"\n"+s2;
return txt;
}
public double arc2area=arc2*circle2.area/2*Math.PI;
public  String derive_Arc2area(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Arc2area=arc2*circle2.area/2*PI";
s2="Arc2area of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(arc2*circle2.area/2*Math.PI);
txt=s1+"\n"+s2;
return txt;
}
public double intersection_area=arc1area+arc2area-triangle1.area-triangle2.area;
public  String derive_Intersection_area(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
s1="Intersection_area=arc1area+arc2area-triangle1.area-triangle2.area";
s2="Intersection_area of "+this.getClass().getName().toLowerCase()+"
"+name+"="+formatter.format(arc1area+arc2area-triangle1.area-triangle2.area);
txt=s1+"\n"+s2;
return txt;
}
Two_circle(Circle h_circle1,Circle h_circle2,Point h_center1,Point h_center2){
circle1=h_circle1;
circle2=h_circle2;
center1=h_center1;
center2=h_center2;
}
public Point  intersection_p1(){
Point p11 =new Point();
p11.x=center1.x+(distance*distance+circle1.radius*circle1.radius-
circle2.radius*circle2.radius)/2*distance;
p11.y=center1.y+Math.sqrt(circle1.radius*circle1.radius- (p11.x-center1.x)*(p11.x-
center1.x));
return p11;
}
public Point  intersection_p2(){
Point p22 =new Point();
p22.x=center2.x+(distance*distance+circle2.radius*circle2.radius-
circle1.radius*circle1.radius)/2*distance;
```

```
p22.y=center2.y+Math.sqrt(circle2.radius*circle2.radius- (p22.x-center2.x)*(p22.x-
center2.x));
return p22;
}
public double  find_area(){
double a ;
a=intersection_area;
return a;
}
public double  find_arc1area(){
double a ;
a=arc1area;
return a;
}
public double  find_arc2area(){
double a ;
a=arc2area;
return a;
}
public double  find_arc1(){
double a ;
a=arc1;
return a;
}
public double  find_arc2(){
double a ;
a=arc2;
return a;
}
public Point  intesection_p1(){
Point a =new Point();
a=intersection_p1();
return a;
}
public Point  intesection_p2(){
Point b =new Point();
b=intersection_p2();
return b;
}
}
```

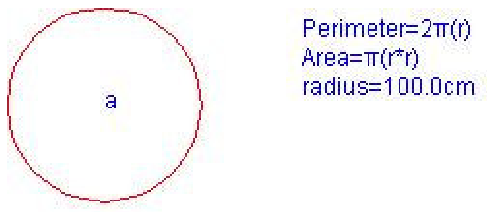
Question 3: given two_circle: a, circle: t1,t2, point: p1,p2;radius of t1=45, x of p1=0, y of p1=0, radius of t2=45, x of p2=60, y of p2=0,circle1 of a=t1, center1 of a=p1,circle2 of a=t2, center2 of a=p2;find arc1 of a, arc2 of a, arc1area of a, arc2area of a,triangle1_area of a,triangle2_area of a, intersection_area of a;

//Generated java code

```
import javax.swing.*;
import java.awt.*;
public class SGMLCircle extends JPanel {
 protected void clear(Graphics g) {
super.paintComponent(g);
}
public static void main(String[] args) {
SGMLDocument document=new SGMLDocument();
document.add_solution_text("Question: given two_circle: a, circle: t1,t2, point:
p1,p2;");
document.add_solution_text("Solution:");
Two_circle      a=new Two_circle();
a.name="a";
Circle   t1=new Circle();
t1.name="t1";
Circle   t2=new Circle();
t2.name="t2";
Point    p1=new Point();
p1.name="p1";
Point    p2=new Point();
p2.name="p2";
t1.setRadius(45);
p1.setLocation(0);
p1.setLocationY(0);
t2.setRadius(45);
p2.setLocation(60);
p2.setLocationY(0);
a.setCircle1(t1);
a.setCenter1(p1);
a.setCircle2(t2);
a.setCenter2(p2);
document.add_solution_text(a.derive_Arc1());
document.add_solution_text(a.derive_Arc2());
document.add_solution_text(a.derive_Arc1area());
document.add_solution_text(a.derive_Arc2area());
document.add_solution_text(a.derive_Triangle1_area());
document.add_solution_text(a.derive_Triangle2_area());
document.add_solution_text(a.derive_Intersection_area());
document.close();
}
}
```

Example 4: Given circle: a; draw a;

PS: PDF output for graphics is supported using hand coded java library for circle.



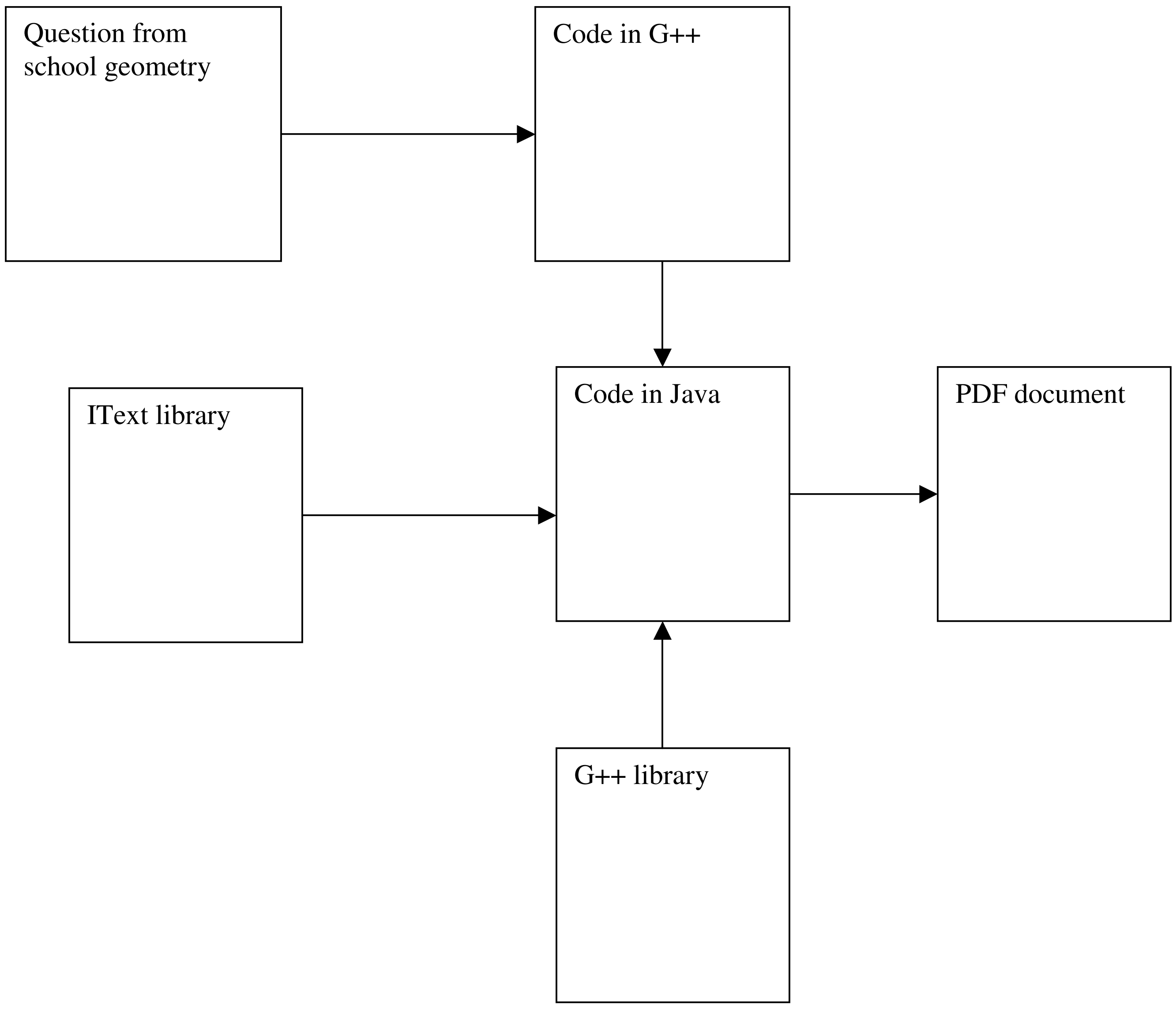
4. Project methodology

Since, it was a team of only one person, it was felt extreme programming will be most appropriate project methodology. Under this method, I followed a very short software release cycle. I tested final output even for a very small change in software. Initial design and code document was not important because the scope of project was changing very rapidly. I trusted myself as a responsible programmer and kept track of all changes as part of informal notes. I used simple zip utility to keep the snapshot of code base every day. Finally, I designed a suite of test to evaluate the software. Intermediate review was done by Professor for acceptability of the project. The complexity in the project was build incrementally. Since, I followed a very flexible and adaptable project methodology, I was able to accommodate even the last minute change request to add new functionality.

Project log

Date	Log
14-oct-2006	Initial proposal
24-oct-2006	Revised proposal with detail syntax of language
30-nov-2006	Decided to add capability to add new shape type and its attribute
15-dec-2006	Final review, test and report

5. Architecture diagram



G++ question generates intermediate java code, which can be linked with iText and G++ library for basic geometric shape. The final output is a PDF document. Usually, it may not be required to recompile G++ library too frequently. However, if the supports for new shapes have been added or attribute/relation for existing shapes have been changed then one needs to recompile g++ library, which will generate new java file. The subsequent run of G++ generated intermediate code for question will take advantage of new functionality provided by G++ library.

6. Test Plan

Below are few test plans used to test the software.

1. Given circle: a; radius of a=10; find perimeter of a, area of a;
2. Given point: a; location of a=(40,50); Find location of a;
3. Given line: ab,point a,b; x of a =10, y of a =10, x of b =100, y of b=100, start_point of ab=a, end_point of ab=b; find length of ab;
4. Given square: a, cube: b; side of a=10, side of b=5; find area of a, volume of b;
5. Given triangle: ABC, Point: A,B,C; x of A=0, y of A=0,x of B=30, y of B=0,x of C=30, y of C=40,a of ABC=A, b of ABC=B, c of ABC=C; Find area of ABC;
6. given point: a, circle: b, triangle: d, square: e; find definition of a, definition of b, definition of d, definition of e;
7. given circle: a; find area of b;
8. given circle: a, point: a; radius of a=5, location of a=(10,10); find area of a;
9. given square: a; radius of a=5; find area of a;
10. given circle: a; draw a; //supported with old version of hand written library

Expected output

1. Question: Given circle: a; radius of a=10; find perimeter of a, area of a;
Solution:

    Perimeter=2*PI*radius
    Perimeter of circle a=62.832
    Area=PI*radius*radius
    Area of circle a=314.159

2.
```
D:\antlr\workspace\anlr>java TestMain
        SGML> Given point: a; location of a=(40,50); Find location_txt of a;
                SGML-003 location is not compatible with point
                SGML-003 location_txt is not compatible with point
```

3. Question: Given line: ab,point: a,b; x of a =10, y of a =10, x of b =100, y of b=100, start_point of ab=a, end_point of ab=b; find length of ab;
Solution:
Length=Point.find_distance(start_point,end_point)
Length of line ab=127.279

4. Question: Given square: a, cube: b; side of a=10, side of b=5; find area of a, volume of b;
Solution:
Area=side*side
Area of square a=100.000
Volume=side*side*side
Volume of cube b=125.000

5. Question: Given triangle: ABC, Point: A,B,C; x of A=0, y of A=0,x of B=30, y of B=0,x of C=30, y of C=40,a of ABC=A, b of ABC=B, c of ABC=C; Find area of ABC;
Solution:
Area=sqrt(S*(S-AB)*(S-BC)*(S-AC))
Area of triangle ABC=600.000

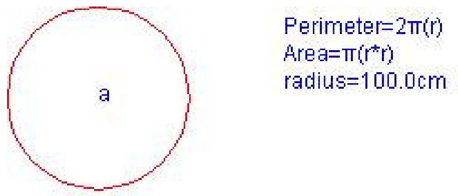
6. Question: given point: a, circle: b, triangle: d, square: e; find definition of a, definition of b,definition of d, definition of e;
Solution:
Definition=a geometric element that has position but no extension; a point is defined by its coordinates
Definition=In Euclidean geometry, a circle is the set of all points in a plane at a fixed distance, called the radius, from a fixed point, called the centre. Circles are simple closed curves, dividing the plane into an interior and exterior. Sometimes the word circle is used to mean the interior, with the circle itself called the circumference. Usually however, the circumference means the length of the circle, and the interior of the circle are called a disk or disc.
Definition=a three-sided polygon
Definition=a plane rectangle with four equal sides and four right angles; a four-sided regular
polygon

7. SGML> given circle: a; find area of b;
SGML-002 shape b is not defined

8. SGML> given circle: a, point: a; radius of a=5, location of a=(10,10); find area of a;
SGML-001 duplicate definition of shape variable a
SGML-003 location is not compatible with circle

9. D:\antlr\workspace\anlr>java TestMain
SGML> given square: a; radius of a=5; find area of a;
SGML-003 radius is not compatible with square

10.



One test was chosen for each geometric shape. I have also shown some of tests, which will give error during semantic test. Specifically, test 8 and test 9 show semantic error handling capability of the language. Also, we took test 10 as a representative example to show the drawing capability of the language. Currently, g++ library does not support generation of graphics but support can be easily added by manually editing generated java code for library. Below is representative java code generated for test case 5

```
import javax.swing.*;
import java.awt.*;
public class SGMLCircle extends JPanel {
protected void clear(Graphics g) {
super.paintComponent(g);
}

public static void main(String[] args) {
SGMLDocument document=new SGMLDocument();
document.add_solution_text("Question: Given triangle: ABC, Point: A,B,C; x of
A=0, y of A=0,x of B=30, y of B=0,x of C=30, y of C=40,a of ABC=A, b of ABC=B,
c of ABC=C; Find area of ABC;");
document.add_solution_text("Solution:");
Triangle        ABC=new Triangle();
ABC.name="ABC";
Point    A=new Point();
A.name="A";
Point    B=new Point();
B.name="B";
Point    C=new Point();
C.name="C";
A.setX(0);
A.setY(0);
B.setX(30);
B.setY(0);
C.setX(30);
C.setY(40);
ABC.setA(A);
ABC.setB(B);
ABC.setC(C);
document.add_solution_text(ABC.derive_Area());
document.close();
}
}

import java.awt.geom.*;
import java.awt.image.BufferedImage;
import java.awt.*;
import java.io.File;
import java.io.IOException;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import javax.imageio.ImageIO;


public class Triangle{
public String name="a";
Triangle(){}
public Point A=new Point(0,0);
public  void setA(Point h_A){
A =h_A;
}

public Point B=new Point(100,0);
public  void setB(Point h_B){
B =h_B;
}
```

```
public Point C=new Point(100,100);
public  void setC(Point h_C){
C =h_C;
}

public String definition="a three-sided polygon";
public  String derive_Definition(){
String txt,s1,s2,s3;
        s1="Definition="+"a three-sided polygon";
        s2="Definition of "+this.getClass().getName().toLowerCase()+"
        "+name+"="+"a three-sided polygon";
        txt=s1+"\n"+s2;
        return s1;
}

public double AB=Point.find_distance(A,B);
public  String derive_Ab(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
        s1="Ab=Point.find_distance(A,B)";
        s2="Ab of "+this.getClass().getName().toLowerCase()+"
        "+name+"="+formatter.format(Point.find_distance(A,B));
        txt=s1+"\n"+s2;
        return txt;
}

public double BC=Point.find_distance(B,C);
public  String derive_Bc(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
        s1="Bc=Point.find_distance(B,C)";
        s2="Bc of "+this.getClass().getName().toLowerCase()+"
        "+name+"="+formatter.format(Point.find_distance(B,C));
        txt=s1+"\n"+s2;
        return txt;
}

public double AC=Point.find_distance(A,C);
public  String derive_Ac(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
        s1="Ac=Point.find_distance(A,C)";
        s2="Ac of "+this.getClass().getName().toLowerCase()+"
        "+name+"="+formatter.format(Point.find_distance(A,C));
        txt=s1+"\n"+s2;
        return txt;
}

public double perimeter=AB+BC+AC;
public  String derive_Perimeter(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
        s1="Perimeter=AB+BC+AC";
        s2="Perimeter of "+this.getClass().getName().toLowerCase()+"
        "+name+"="+formatter.format(AB+BC+AC);
        txt=s1+"\n"+s2;
        return txt;
}

public double S=perimeter/2;
public  String derive_S(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
        s1="S=perimeter/2";
        s2="S of "+this.getClass().getName().toLowerCase()+"
        "+name+"="+formatter.format(perimeter/2);
        txt=s1+"\n"+s2;
        return txt;
}

public String theorem_pythagorean="Proof #1:This is probably the most famous
of all proofs of the Pythagorean proposition. It's the first of Euclid's two proofs
(I.47). The underlying configuration became known under a variety of names, the
Bride's Chair likely being the most popular. The proof has been illustrated by an
award winning Java applet written by Jim Morey. I include it on a separate page
with Jim's kind permission. The proof below is a somewhat shortened version of
the original Euclidean proof as it appears in Sir Thomas Heath's translation.First
of all, ?ABF = ?AEC by SAS. This is because, AE = AB, AF = AC, and BAF =  BAC
+  CAF =  CAB +  BAE =  CAE.      ?ABF has base AF and the altitude from B
equal to AC. Its area therefore equals half that of square on the side AC. On the
```

```
other hand, ?AEC has AE and the altitude from C equal to AM, where M is the
point of intersection of AB with the line CL parallel to AE. Thus the area of ?AEC
equals half that of the rectangle AELM. Which says that the area AC2 of the
square on side AC equals the area of the rectangle AELM.Similarly, the are BC2
of the square on side BC equals that of rectangle BMLD. Finally, the two
rectangles AELM and BMLD make up the square on the hypotenuse AB.";

public  String derive_Theorem_pythagorean(){
String txt,s1,s2,s3;

s1="Theorem_pythagorean="+"Proof #1:This is probably the most famous of all
proofs of the Pythagorean proposition. It's the first of Euclid's two proofs (I.47).
The underlying configuration became known under a variety of names, the
Bride's Chair likely being the most popular. The proof has been illustrated by an
award winning Java applet written by Jim Morey. I include it on a separate page
with Jim's kind permission. The proof below is a somewhat shortened version of
the original Euclidean proof as it appears in Sir Thomas Heath's translation.First
of all, ?ABF = ?AEC by SAS. This is because, AE = AB, AF = AC, and BAF =  BAC
+  CAF =  CAB +  BAE =  CAE.      ?ABF has base AF and the altitude from B
equal to AC. Its area therefore equals half that of square on the side AC. On the
other hand, ?AEC has AE and the altitude from C equal to AM, where M is the
point of intersection of AB with the line CL parallel to AE. Thus the area of ?AEC
equals half that of the rectangle AELM. Which says that the area AC2 of the
square on side AC equals the area of the rectangle AELM.Similarly, the are BC2
of the square on side BC equals that of rectangle BMLD. Finally, the two
rectangles AELM and BMLD make up the square on the hypotenuse AB.";
s2="Theorem_pythagorean of "+this.getClass().getName().toLowerCase()+"
"+name+"="+"Proof #1:This is probably the most famous of all proofs of the
Pythagorean proposition. It's the first of Euclid's two proofs (I.47). The
underlying configuration became known under a variety of names, the Bride's
Chair likely being the most popular. The proof has been illustrated by an award
winning Java applet written by Jim Morey. I include it on a separate page with
Jim's kind permission. The proof below is a somewhat shortened version of the
original Euclidean proof as it appears in Sir Thomas Heath's translation.First of
all, ?ABF = ?AEC by SAS. This is because, AE = AB, AF = AC, and BAF =  BAC +
CAF =  CAB +  BAE =  CAE.          ?ABF has base AF and the altitude from B
equal to AC. Its area therefore equals half that of square on the side AC. On the
other hand, ?AEC has AE and the altitude from C equal to AM, where M is the
point of intersection of AB with the line CL parallel to AE. Thus the area of ?AEC
equals half that of the rectangle AELM. Which says that the area AC2 of the
square on side AC equals the area of the rectangle AELM.Similarly, the are BC2
of the square on side BC equals that of rectangle BMLD. Finally, the two
rectangles AELM and BMLD make up the square on the hypotenuse AB.";
```

```
        txt=s1+"\n"+s2;
        return s1;
}

public double area=Math.sqrt(S*(S-AB)*(S-BC)*(S-AC));
public  String derive_Area(){
String txt,s1,s2,s3;
NumberFormat formatter = new DecimalFormat("000000");
formatter = new DecimalFormat("0.000");
        s1="Area=sqrt(S*(S-AB)*(S-BC)*(S-AC))";
        s2="Area of "+this.getClass().getName().toLowerCase()+"
        "+name+"="+formatter.format(Math.sqrt(S*(S-AB)*(S-BC)*(S-AC)));
        txt=s1+"\n"+s2;
        return txt;
}

Triangle(Point h_A,Point h_B,Point h_C){
A=h_A;
B=h_B;
C=h_C;
}

public double  find_angle(){
double h_area ;
h_area=area;
return h_area;
}
}
```

7. Lesson learned

a. Scope of the project The initial scope of the project was too ambitious. As I started working through the project, I realized that solving or even attempting to describe geometry problem in natural language is a very hard problem. Even using context free grammar, it is still hard to solve high school geometry using computer language. Below are list of features, which were planned but not implemented because it was either too difficult or due time/resource constraint.

b. Parsing natural language question: Originally, I had planned to parse natural language question using regular expression and pattern matching. It would have been too difficult and was beyond the scope of this course.

c. Solving question algorithmically: My initial thought was that one can derive solution from given facts. I was again wrong here. It is a hard problem to generate solution automatically. For example: If side of cube is given one can derive volume of cube given the formula for volume. However, if volume is given then one can't automatically derive side unless a formula for side is given in terms of volume. Thus, code generation to solve equation of unknown variable is not automatic and it is a hard problem.

d. Graphics layout generation: I was tempted to include geometric construction into language. It was okay for a simple graphics for few geometric shapes. However, as number increased and it became messy to optimize the page layout to place all objects in one page. This is a real optimization problem with concept of Human Computer Interaction. It could be a project on its own.

Advice for future team: Changing the scope of project at last moment may be difficult to adopt in project. Visualize many examples and try to get approval about the full scope of project before you can begin coding for it. Don't hesitate to rewrite the complete grammar. The rewriting complete grammar may be easier than doing a patch-work to existing one.

8. Appendix

Code listing

Name	**Comment**
Lib.bat	Batch file to compile g++ library
Sgml.bat	Batch file to compile g++ question
TestLexer.g	Lexer file for g++ question
TestWalker.g	Walker file
LibLexer.g	Lexer file for g++ library
LibWaker.g	Walker file for g++ library
ID.java	Variable id
Util.java	Utility class to support other class
SymbolTable.java	Resolving variable name and scope
Type.java	Shape Type
SGMLDocument.java	Wrapper class for iText
TestMain.java	Main driving program for question g++
LibMain.java	Main driving program for library g++

```
/*******************************************************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: TestLexer.g
*******************************************************************************************************/


class TestParser extends Parser;
        options {
                        buildAST = true; // Enable AST building
                        k = 3;        // Need to distinguish between          ID by
                        itself and ID ASSIGN
                        exportVocab = TestParser;
                                        //caseSensitive=false ;
         }



        tokens
         {  //ID;
                     DECLS;
                     VAR_LIST;
                     FACT_LIST;
                     FACT;
                     FACT1;
                     ASK;                //DRAW;
                     QUESTION;

               }

question : decls statements (ask_question|draw_object)
//{#question=(#[QUESTION,"QUESTION"],question);}
;
ask_question : FIND! ask ( COMMA! ask)* SEMI!
{#ask_question=#([ASK,"ASK"],#ask_question);}
|{#ask_question=#([ASK,"ASK"],ask_question);}
;
draw_object: DRAW^ var_list SEMI!
//{#draw_object=#([DRAW,"DRAW"],#draw_object);}
//|{#draw_object=#([DRAW,"DRAW"],draw_object);}
;
//ask : ATTRIBUTE^ "of"! ID;
ask : attribute "of"! ID;
```

```
attribute: ID;
/*template:     "distance" "between"! ("point")?! ID ("and"|COMMA)?! ID
             |ANGLE   BETWEEN  LINE  ID AND ID
             |  AREA  (options {greedy=true;}: OF INTERSECTION BETWEEN
CIRCLE ID AND ID
             |  AREA  OF CIRCLE INSCRIBED WITHIN TRAINGLE ID);
*/
   //   |("perimeter"^ "of"! ID)
//attribute : "area"|"perimeter";

decls : GIVEN! object_def (COMMA! object_def)* SEMI!
{ #decls =#([DECLS,"DECLS"], #decls);}
|{ #decls =#([DECLS,"DECLS"], #decls);}
;

object_def : object_type COLON! var_list  ;
object_type : ID;
var_list :  ID (COMMA! ID)*
{#var_list=#([VAR_LIST,"VAR_LIST"],var_list);}
|{#var_list=#([VAR_LIST,"VAR_LIST"],var_list);}
;

statements: (assignment)(COMMA! assignment)* SEMI!
{ #statements =#([FACT_LIST,"FACT_LIST"], #statements);}
|{ #statements =#([FACT_LIST,"FACT_LIST"], #statements);}
;
assignment: attr_name "of"! lvalue
{ #assignment =#([FACT,"FACT"], #assignment);}
//|{ #assignment =#([FACT1,"FACT1"], #assignment);}
;
attr_name: ID;
lvalue: ID ASSIGN! value
//{ #lvalue =#([LVALUE,"LVALUE"], #lvalue);}
;

value: NUMBER
      | LBRAC! NUMBER COMMA! NUMBER RBRAC!
      | ID
      ;
```

```
class TestLexer extends Lexer;
                  options {
                                    testLiterals = false; // By default, don't check tokens
against keywords
   k = 3;              // Need to decide when strings literals end
   charVocabulary = '\3'..'\377'; // Accept all eight-bit ASCII characters
caseSensitive=false ;
                             }
SEMI        : ';';
FULLSTOP     : '.';
COMMA       : ',';
ASSIGN      : '=';
LBRAC       : '(';
RBRAC       : ')';
COLON        : ':';
GIVEN       : "given";
FIND        : "find";
DRAW        : "draw";
protected LETTER : ( 'a'..'z'  ) ;
protected DIGIT  : '0'..'9' ;
protected
Exponent
: ('e') ('+'|'-')? ('0'..'9')+
;
NUMBER: ('0'..'9')+
( '.' ('0'..'9')* (Exponent)?)?|( '.' ('0'..'9')* (Exponent)?)
;

ID
         options {
                           testLiterals = true;
                    }
   : LETTER (LETTER | DIGIT | '_')* ;
WS  :   ( ' '
      | '\t'
      | '\n' { newline(); }
      | '\r'
      ) { $setType(Token.SKIP); }
   ;
```

```
/************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: TestWalker.g
************************************************************/

            {
import javax.swing.*; // For JPanel, etc.
import java.awt.*;
import java.io.*;
}


class TestWalker extends TreeParser;
            options {
                        importVocab=TestParser;
                        //k = 2;
                    }           {
                        SymbolTable st=new SymbolTable();
                    }
                    question
{
             Type t=null;
             String s="";
  try {
                PrintStream out =new PrintStream(new
                FileOutputStream("c:\\SGMLCircle.java"));
  try {
                BufferedReader in = new BufferedReader(new
                FileReader("c:\\question.txt"));
            String str;
            while ((str = in.readLine()) != null) {
            s=s+str;
        }
            in.close();
             }
  catch (IOException e) {
             }

              out.print("import javax.swing.*; \n");
        out.print("import java.awt.*;\n");
        out.print("public class SGMLCircle extends JPanel {\n");
        out.print(" protected void clear(Graphics g) {\n");
        out.print("super.paintComponent(g);\n}\n");
        out.print("public static void main(String[] args) {\n");
             out.print("SGMLDocument document=
                   new SGMLDocument();\n");
                out.print("document.add_solution_text(\"Question: "+s+"\");\n");
                 out.print("document.add_solution_text("+"\"Solution:\""+");\n");
         out.close();

            }
    catch (Exception ex) { ex.printStackTrace();}

       }

: decls  statements (ask_question | draw_object)
{

   // WindowUtilities.openInJFrame(new ShapeExample(), 580, 400);
  //  Document document = new Document(PageSize.A4, 50, 50, 50, 50);
   try {
                   PrintStream out =new PrintStream(new
                   FileOutputStream("c:\\SGMLCircle.java",true));

              out.print("document.close();\n");
                   out.print("}\t\n");
                   out.print("}\n");
                   out.close();
                   Runtime rt=Runtime.getRuntime();
          String command = "c:\\sgml\\sgml";
          String shell   = "cmd.exe";
          String opt     = "/c";
          String[] cmdArray = {shell, opt, command};

                   Process pp= rt.getRuntime().exec(cmdArray);

          }
          catch (Exception ex) {ex.printStackTrace();}

          }
;
draw_object
{
}
: #(DRAW #(VAR_LIST (ID   {
        try {
                PrintStream out =new PrintStream(new
                FileOutputStream("c:\\SGMLCircle.java",true));

                out.print(#ID.getText()+".draw(document.g2dd);\n");
               out.close();
          }
        catch (Exception ex) {ex.printStackTrace();}

          }
       )*{
     try {
                    PrintStream out =new PrintStream(new
                    FileOutputStream("c:\\SGMLCircle.java",true));


                    out.print("document.add_solution_drawing("+"\"c:\\"+"\\
                    circle.jpg\""+");\n");
             out.close();
        }
     catch (Exception ex) {ex.printStackTrace();}
        }))
;

ask_question
{
        Type t=null;
}
:  #(ASK ((a:ID ID  {
    Id tt=st.get(#ID.getText());
    if (tt==null)
                System.out.print("SGML-002 shape "+#ID.getText()+" is not
                defined\n");
                else if (!tt.getType().attrMap(a.getText()))
                System.out.print("SGML-003 "+ a.getText()+ " is not compatible
                with "+tt.getType().name+"\n");


              try {
                    PrintStream out =new PrintStream(new
                    FileOutputStream("c:\\SGMLCircle.java",true));

                    out.print("document.add_solution_text(");
                    out.print(#ID.getText()+".derive_"+Util.initCap(a.getText())+
                    "());\n");
                    out.close();



                     }

                catch (Exception ex) {ex.printStackTrace();}

                     }
               ) )*

               )
;

decls
{ Type t=null;
      //SybmolTable top;

}
: #(DECLS  (object_def  {System.out.print("\n");}
              )*
 );

object_def
 {
      //Type t=null;
      //Type t=null;
 }
 :(v:ID
 //ID (ID {System.out.print(" "+  #ID.getText()+"=new "+v.getText()+"();\n");})*
      #(VAR_LIST (t:ID  {

               Id id=st.get(t.getText());
               if (id ==null)
                    st.put(t.getText(),new Type(v.getText(),100));
               else
                    System.out.print("SGML-001 duplicate
                    definition of shape variable "+t.getText()+"\n");
               try {
```

```
                            PrintStream out =new PrintStream(new
                            FileOutputStream("c:\\SGMLCircle.java",true));

                        out.print(Util.initCap(v.getText())+"\t");
                            out.print(" "+  t.getText()+"=new
                            "+Util.initCap(v.getText())+"();\n");

                            out.print(t.getText()+".name="+"\""+t.getText()+"\""+";\n");
                            out.close();
                        }
                      catch (Exception ex) {ex.printStackTrace();}

                        })*)
                        )

            ;
             statements
             {
                  int i;
                  i=0;
             }
              :#(FACT_LIST   (assignment {
                           try {
                                   PrintStream out =new PrintStream(new
                                   FileOutputStream("c:\\SGMLCircle.java",true));

                                 System.out.print("\n");
                                }
                           catch (Exception ex) {ex.printStackTrace();}
                         })*)
              ;
             assignment
             {
                  int i;
                  i=0;
             }
              :#(FACT (att1:ID

                  it:ID
                     {
                        Id tt=st.get(it.getText());
                        if (tt==null)
```

```
                                        System.out.print("SGML-002 shape
                                        "+it.getText()+" is not defined\n");
                                        else if (!tt.getType().attrMap(att1.getText()))
                                        System.out.print("SGML-003 "+ att1.getText()+
                                        " is not compatible with
                                        "+tt.getType().name+"\n");

                        }


   value[it.getText(),att1.getText()]


     ))



 ;
   value [String it,String att1]
   {
   }
:ID  {

                        try {
                              PrintStream out =new PrintStream(new
                              FileOutputStream("c:\\SGMLCircle.java",true));

                              out.print(it+".set"+Util.initCap(att1)+"("+#ID.getText()+");\
                              n");
                          }
                         catch (Exception ex) {ex.printStackTrace();}

                      }
 |id1:NUMBER
 {

                         try {
                               PrintStream out =new PrintStream(new
                               FileOutputStream("c:\\SGMLCircle.java",true));
                        out.print(it+".set"+Util.initCap(att1)+"("+id1.getText()+");\n");
                             }
                         catch (Exception ex) {ex.printStackTrace();}

                             }
```

```
  (id2:NUMBER{

                    try {
                              PrintStream out =new PrintStream(new
                              FileOutputStream("c:\\SGMLCircle.java",true));

                              out.print(it+".set"+Util.initCap(att1)+"Y("+id2.getText()+")
                              ;\n");
                            out.close();
                          }
                    catch (Exception ex) {ex.printStackTrace();}

                          })?
        ;
  lvalue
 {
        int i;
        i=0;
 }
  : #( NUMBER {System.out.print("number:\n");})
  ;


/********************************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: SGMLDocument.java
********************************************************************************/


import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.Paragraph;
import com.lowagie.text.pdf.PdfWriter;
import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.Toolkit;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.io.*;
import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.Font;
import com.lowagie.text.Image;
import com.lowagie.text.Phrase;
import com.lowagie.text.pdf.BaseFont;
import com.lowagie.text.pdf.PdfContentByte;
import com.lowagie.text.pdf.PdfWriter;
import java.awt.image.*;

import javax.imageio.ImageIO;

class SGMLDocument
{
        private int total_page; // total number of pages
        private int positionX;  //cursor x position
        private int positionY;  //cursor y position
        private int width;      // width of document
        private int height;     // height of document
        private int img_height; // image height
        private int img_width;  // image width
        private Document document; // solution document
        private  PdfWriter solution;
        private java.awt.Image figure; // drawing
        private BufferedImage bufferedImage;
        private int image_width;
        private int image_height;
        public Graphics2D g2dd ;



 public  void init_graphics(){
//   try {
        bufferedImage = new BufferedImage(image_width,       image_height/2,
        BufferedImage.TYPE_INT_RGB);
   // Create a graphics contents on the buffered image
       g2dd = bufferedImage.createGraphics();
       g2dd.setBackground(Color.WHITE);
       g2dd.fillRect(0, 0, image_width, image_height);

     }
  public void save_graphics(){
    try{
          File file = new File("c:\\circle.jpg");
          ImageIO.write(bufferedImage, "jpg", file);

      } catch (IOException e) {
  }
  }

SGMLDocument(){

try {


         document = new Document();
         // step 2:
         // we create a writer that listens to the document
         // and directs a PDF-stream to a file

         solution=PdfWriter.getInstance(document,`F new
         FileOutputStream("c:\\HelloWorld2.pdf"));

         image_width = (int)solution.getPageSize().width();

         image_height = (int)solution.getPageSize().height();

         init_graphics();

         width=(int)solution.getPageSize().width();

         height=(int)solution.getPageSize().height();
         // step 3: we open the document
    document.open();

  }
catch(DocumentException de) {
    System.err.println(de.getMessage());
}
catch(IOException ioe) {

    System.err.println(ioe.getMessage());
}
}
```

```
public void close()
{
     //save_graphics();

       document.close();
}
 public  void add_solution_text(String text){

           System.setProperty("file.encoding", "UTF-8");
try {

           BaseFont helvetica = BaseFont.createFont(BaseFont.HELVETICA,
           BaseFont.CP1252, BaseFont.EMBEDDED);
    Font font = new Font(helvetica, 12, Font.NORMAL);
    NumberFormat formatter = new DecimalFormat("000000");
    formatter = new DecimalFormat("0.000");

           document.add(new Paragraph(text));
             // PdfContentByte cb = solution.getDirectContent();
}

catch(DocumentException de) {

    System.err.println(de.getMessage());

    }
catch(IOException ioe) {
    System.err.println(ioe.getMessage()

    }
    }
 public void add_solution_drawing(String fname) {

 try {

           save_graphics();
    figure = Toolkit.getDefaultToolkit().createImage(fname);
    Image image = Image.getInstance(figure, null);
           PdfContentByte cb = solution.getDirectContent();
    image.setAbsolutePosition(10,10);
           cb.addImage(image);
    }
catch(DocumentException de) {
    System.err.println(de.getMessage());
    }
catch(IOException ioe) {
    System.err.println(ioe.getMessage());
    }
    }
}
```

```
/*************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: SGMLDataType.java
*************************************************************/
```

```
import java.io.PrintWriter;
public class SGMLDataType
{
   String name;   // used in hash table

   public SGMLDataType() {
      name = null;
   }

   public SGMLDataType( String name ) {
      this.name = name;
   }

   public String typename() {
      return "unknown";
   }

   public SGMLDataType copy() {
      return new SGMLDataType();
   }

   public void setName( String name ) {
      this.name = name;
   }
```

```
public SGMLDataType error( String msg ) {
   throw new SGMLException( "illegal operation: " + msg
                     + "( <" + typename() + "> "
                     + ( name != null ? name : "<?>" )
                     + " )" );
}
public SGMLDataType error( SGMLDataType b, String msg ) {
   if ( null == b )
      return error( msg );
   throw new SGMLException(
      "illegal operation: " + msg
      + "( <" + typename() + "> "
      + ( name != null ? name : "<?>" )
      + " and "
      + "<" + typename() + "> "
      + ( name != null ? name : "<?>" )
      + " )" );
}

public void print( PrintWriter w ) {
   if ( name != null )
      w.print( name + " = " );
   w.println( "<undefined>" );
}

public void print() {
   print( new PrintWriter( System.out, true ) );
}

public void what( PrintWriter w ) {
   w.print( "<" + typename() + ">  " );
   print( w );
}

public void what() {
   what( new PrintWriter( System.out, true ) );
}

public SGMLDataType assign( SGMLDataType b ) {
   return error( b, "=" );
}
```

```
   public SGMLDataType uminus() {
      return error( "-" );
   }


   public SGMLDataType plus( SGMLDataType b ) {
      return error( b, "+" );
   }

   public SGMLDataType add( SGMLDataType b ) {
      return error( b, "+=" );
   }

   public SGMLDataType minus( SGMLDataType b ) {
      return error( b, "-" );
   }


}

/***************************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: SGMLException.java
***************************************************************************/

class SGMLException extends RuntimeException {
   SGMLException( String msg ) {
      System.err.println( "Error: " + msg );
   }
}


/***************************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: SGMLSymbolTable.java
***************************************************************************/

import java.util.*;
import java.io.PrintWriter;
```

```
//class SGMLSymbolTable extends HashMap {
public class SymbolTable {
private Hashtable table;
        SymbolTable() {
        table=new Hashtable();}
    public void put (String token, Type t){
    table.put(token, new Id(token,t));
}

public Id get(String token){
  for ( Enumeration e = table.keys() ; e.hasMoreElements() ; )
  {
    Id id=(Id)(table.get(token));
    String key = (String)e.nextElement();
    //System.out.print("id");
    if (id!=null) return id;
  }
  return null;

}
}
/*****************************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: Util.java
*****************************************************************************/

class Util {

        public static String initCap (String in)
        {
                if (in == null || in.length() == 0)
                        return in;
                //
                boolean capitalize = true;
                char[] data = in.toCharArray();
                for (int i = 0; i < data.length; i++)
                {
                                if (data[i] == ' ' || Character.isWhitespace(data[i]))
                                capitalize = true;
                        else if (capitalize)
                        {
                                data[i] = Character.toUpperCase (data[i]);
                        capitalize = false;
                    }
                    else
                        data[i] = Character.toLowerCase (data[i]);
                }
                return new String (data);
        }       //      initCap



}
```

/***
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: Type.java
***/

```
import java.io.*;
import java.util.*;

public class Type {
public int width=0;
public String name ="";
public Type(String s,int w) {name=s; width=w;}
public static final Type
Circle =new Type("circle",100),
Line =new Type ("line",100),
Cube =new Type("cube",100);
public boolean attrMap(String attr)
{
   try { // Create a FileReader and then wrap it with BufferedReader.

                FileReader file_reader = new FileReader (new
                File("c:\\sgml\\attrmap.txt"));
                BufferedReader buf_reader = new BufferedReader (file_reader); //
                Read each line of the file and look for the string of interest.
     do {
        String line = buf_reader.readLine ();
                 if (line == null) break;
                    String s[]=line.split(":");
                 if (name.equalsIgnoreCase(s[0]))
                                                                        {
```

```
                                                                  if
                         (s[1].indexOf(attr)!=-1) {

                         return true;
                                                                }
    }
  }

 while (true);

        buf_reader.close ();
 }
 catch (IOException e) {
     System.out.println ("IO exception =" + e );
 }
     return false;
 }
 }
/**************************************************************
```

Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Question g++
File: TestMain.java

```
**************************************************************/

import java.io.*;
import antlr.ByteBuffer;
import antlr.CommonAST;
import antlr.collections.AST;
import antlr.RecognitionException;
import antlr.TokenStreamException;
import antlr.TokenStreamIOException;
import antlr.Token;

class TestMain {
   static boolean verbose = true;

   public static void execFile( String filename ) {
      try
      {
         InputStream input = ( null != filename ) ?
            (InputStream) new FileInputStream( filename ) :
            (InputStream) System.in;
```

```
    TestLexer lexer = new TestLexer( input );

    TestParser parser = new TestParser( lexer );
    parser.question();

    CommonAST tree = (CommonAST)parser.getAST();

    if ( verbose )
    {
       // Print the resulting tree out in LISP notation
       System.out.println(
                    "================ tree structure
                    ======================" );
       System.out.println( tree.toStringList() );
    }



    TestWalker walker = new TestWalker();
    // Traverse the tree created by the parser

    if ( verbose )
       System.out.println(
                    "================ program output
                    ======================" );
    walker.question(tree);


    if ( verbose )
       System.out.println(
                    "================ program return
                    ======================" );


 } catch( IOException e ) {
    System.err.println( "Error: I/O: " + e );
 } catch( RecognitionException e ) {
    System.err.println( "Error: Recognition: " + e );
 } catch( TokenStreamException e ) {
    System.err.println( "Error: Token stream: " + e );
 } catch( Exception e ) {
    System.err.println( "Error: " + e );
 }
```

```
}

public static void commandLine() {
        InputStream input = (InputStream)
        new DataInputStream( System.in );
   TestWalker walker = new TestWalker();
   System.out.print( "SGML> " );
   System.out.flush();

   try{
                        PrintStream out =new PrintStream(new
                        FileOutputStream("c:\\question.txt"));

         input.mark(0);
                        for(int i = input.read(); i != -1; i =     input.read())
         {
            out.print((char)i);
            if (i==10) break;
         }

         input.reset();
         out.close();
    }
   catch (Exception e) {}

        TestLexer lexer = new TestLexer( input );
        TestParser parser = new TestParser( lexer );



        try
          {
            parser.question();
            CommonAST tree = (CommonAST)parser.getAST();
            walker.question(tree);

          }
       catch( RecognitionException e ) {
                              System.err.println( "Recognition exception: "  + e );
          }
       catch( TokenStreamException e ) {
```

```
                if ( e instanceof TokenStreamIOException ) {
                System.err.println( "Token I/O exception" );

              }
                                  System.err.println( "Error: Token stream: " + e );
              }
           catch( RuntimeException e ) {
               System.err.println( "Error: Runtime: " + e );
               e.printStackTrace();
              }
            catch( Exception e ) {
                System.err.println( "Error: " + e );
                e.printStackTrace();
              }


    }
  public static void main( String[] args ) {

    verbose = args.length >= 1 && args[0].equals( "-v" );
    boolean batch = args.length >=1 && args[0].equals( "-b" );

    if ( args.length >= 1 && args[args.length-1].charAt(0) != '-' )
           execFile( args[args.length-1] );
        else if ( batch )
           execFile( null );
        else
           commandLine();

        System.exit( 0 );
    }
  }


/*************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Module: Library g++
Date: 15-Dec-2006
File: LibLexer.g
*************************************************************/


class LibParser extends Parser;
options {
    buildAST = true; // Enable AST building
    k = 2; / Need to distinguish between ID by itself and ID ASSIGN
   exportVocab = LibParser;
}

 tokens {

FACT;
FACT_LIST;
RELATION_LIST;
DEFINTION;

}


definition: object
            (attribute|derived_attribute)+
             relations SEMI!;
object: SHAPE^ COLON! ID SEMI!;
derived_attribute: DERIVED ATTRIBUTE! COLON! ID COMMA!
            TYPE! object_type COMMA!
            (DEFAULT! def_value)
            (COMMA! FORMAT!  STRING )? SEMI!;
attribute: ATTRIBUTE! COLON! ID COMMA!
            TYPE! object_type COMMA!
            (DEFAULT! def_value)
            (COMMA! FORMAT!  STRING )? SEMI!;
def_value: NUMBER
            |LBRAC NUMBER (COMMA NUMBER)* RBRAC!
            |LBRAC! ID (COMMA ID)* RBRAC!
            |STRING
            |EXPR LBRAC! STRING RBRAC!
           ;
relations: (relation1|relation2) (SEMI! (relation1|relation2))*
{ #relations =#([RELATION_LIST,"RELATION_LIST"], #relations);}
|{ #relations =#([RELATION_LIST,"RELATION_LIST"], #relations);}
;
relation1: relation_name (input_type)? (output_type)? facts;
relation2: GLOBAL relation_name (input_type)? (output_type)? facts;
relation_name: RELATION COLON! ID;
input_type: REFERS object_type ID (COMMA! object_type ID)*;
output_type: RETURNS object_type ID;
facts: FORMULA statements;
```

```
//RELATION^ COLON! relation_name REFERS object_type value
//                      RETURNS object_type
 //                     FORMULA^ statements;
//relation_name: ID;
object_type: NUM|TEXT|ID;
var_list: ID (COMMA ID)*;
statements: statement (SEMI! statement)*
{ #statements =#([FACT_LIST,"FACT_LIST"], #statements);}
|{ #statements =#([FACT_LIST,"FACT_LIST"], #statements);}
;
statement: lvalue ASSIGN rvalue;
lvalue: ID (DOT ID)*;

rvalue:   value  (OPERATOR value)*  |EXPR LBRAC! STRING RBRAC!
        ;
value: CONSTANT|ID|STRING|NUMBER;




class LibLexer extends Lexer;
options {
   testLiterals = false; // By default, don't check tokens against keywords
   k = 5;              // Need to decide when strings literals end
   charVocabulary = '\3'..'\377'; // Accept all eight-bit ASCII characters
caseSensitive=false ;

}

SEMI         :          ';';
PLUS         :         '+';
DOT          :         '.';
COMMA      :         ',';
ASSIGN       :        '=';
LBRAC        :        '(';
RBRAC         :       ')';
COLON        :        ':';
SHAPE         :       "shape";
EXPR          :        "expr";
ATTRIBUTE :         "attribute";
```

```
TYPE          :         "type";
DERIVED       :     "derived";
GLOBAL        :     "global";
DEFAULT       :     "default";
FORMAT        :     "format";
RELATION      :     "relation";
REFERS        :     "refers";
RETURNS       :     "returns";
FORMULA       :     "formula";
TEXT          :         "text";
NUM           :         "number";
//CONSTANT: "pi"|"e";

protected LETTER : ( 'a'..'z'  ) ;
protected DIGIT  : '0'..'9' ;
//NUMBER       : (DIGIT)+;
protected
Exponent       : ('e') ('+'|'-')? ('0'..'9')+
;
NUMBER         : ('0'..'9')+ ( '.' ('0'..'9')* (Exponent)?)?|( '.' ('0'..'9')* (Exponent)?);

OPERATOR       : '+'|'-'|'*'|'/'|'^';
STRING         : '"'!
           ( ~('"' | '\n')| ('"'!'"'))* '"'! ;
CONSTANT       : "pi";

ID
options {
   testLiterals = true;
}
   : LETTER (LETTER | DIGIT | '_')* ;
WS  :   ( ' '
       | '\t'
       | '\n' { newline(); }
       | '\r'
      ) { $setType(Token.SKIP); }
   ;

/***************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Library g++
File: LibWalker.g
***************************************************************/
```

```
{
import javax.swing.*; // For JPanel, etc.
import java.awt.*;
import java.io.*;

}

class LibWalker extends TreeParser;
options {
importVocab=LibParser;
}

definition
{
        String s1,s,object_name;
        s="";
        s1="";
        object_name="";

   try {
                PrintStream out =new PrintStream(new
                FileOutputStream("c:\\temp.java"));

                out.print("import java.awt.geom.*; \n");
                out.print("import java.awt.image.BufferedImage; \n");
                out.print("import java.awt.*; \n");
                out.print("import java.io.File; \n");
                out.print("import java.io.IOException; \n");
                out.print("import java.text.DecimalFormat; \n");
                out.print("import java.text.NumberFormat; \n");
                out.print("import javax.imageio.ImageIO; \n");
                out.close();

     }
   catch (Exception ex) { ex.printStackTrace();}
     }
: object_name=object
(s=attribute { s1=s1+s+",";}|derived_attribute )+
{
        try {
                 PrintStream out =new PrintStream(new
                 FileOutputStream("c:\\temp.java",true));
```

```
                    out.print(Util.initCap(object_name)+"("+s1.substring(0,s1.length()-
                    1)+"){\n"+Util.parse(s1));
                   out.print("}\n");
              }
      catch (Exception ex) { ex.printStackTrace();}
         }
                    relations {
          try {
                  PrintStream out =new PrintStream(new
                  FileOutputStream("c:\\temp.java",true));

            out.print("}\n");
            out.close();
            Runtime rt=Runtime.getRuntime();
              String command = "copy c:\\temp.java
              c:\\sgml\\lib\\"+Util.initCap(object_name)+".java";
        String shell   = "cmd.exe";
        String opt     = "/c";
        String[] cmdArray = {shell, opt, command};

             Process pp= rt.getRuntime().exec(cmdArray);
           }
      catch (Exception ex) { ex.printStackTrace();}
           }
;


object returns [String s]
{
        s="";
}
: #(SHAPE ID {s=#ID.getText();}

        {
                try {
                        PrintStream out =new PrintStream(new
                        FileOutputStream("c:\\temp.java",true));

                        out.print("\n public class    "+Util.initCap(#ID.getText())+
                        "{\n"+"public String name=\"a\";\n"
                        +Util.initCap(#ID.getText())+ "("+
                       "){}\n");
                    }
          catch (Exception ex) { ex.printStackTrace();}
                     }) ;

relations
{
}
:#(RELATION_LIST (relation1|relation2)  (relation1|relation2)* )
;
relation1
{
          String s,t1;
          Type t;
          t1="void";
          String input="";
}
:  (s=relation_name (input=input_type)? (t1=output_type)?
{
          try {
                      PrintStream out =new PrintStream(new
                      FileOutputStream("c:\\temp.java",true));

                      if (t1.split(" ")[0].equals("double"))
                                   out.print("public "+t1.split(" ")[0] +"
                                   "+s+"("+input+"){\n"+ t1+ " ;");
                    else
                              out.print("public "+t1.split(" ")[0] +"  "+s+"("+input+"){\n");
                    if (!t1.equals("void")&&!t1.split(" ")[0].equals("double"))
                              out.print( t1+ " =new "+t1.split(" ")[0]+"();");
    }
        catch (Exception ex) { ex.printStackTrace();}

 }
 facts {
                    try {
                                PrintStream out =new PrintStream(new
                                FileOutputStream("c:\\temp.java",true));
                               if (!t1.equals("void"))
                    out.print("return "+t1.split(" ")[1]+";\n");
                    out.print("}\n");
                  }
        catch (Exception ex) { ex.printStackTrace();}
```

```
                        }
 )

;
relation2
{
        String s,t1;
        Type t;
        t1="void";
        String input="";
}
: GLOBAL (s=relation_name (input=input_type)? (t1=output_type)?
{
        try {
                    PrintStream out =new PrintStream(new
                    FileOutputStream("c:\\temp.java",true));

                    if (t1.split(" ")[0].equals("double"))
                    out.print("public static "+t1.split(" ")[0] +"  "+s+"("+input+"){\n"+
                    t1+ " ;");
         else
                      out.print("public static "+t1.split("    ")[0] +"
                    "+s+"("+input+"){\n");
                    if (!t1.equals("void")&&!t1.split(" ")[0].equals("double"))
         out.print( t1+ " =new "+t1.split(" ")[0]+"();");
      }
    catch (Exception ex) { ex.printStackTrace();}

      }
 facts  {

        try {
                    PrintStream out =new PrintStream(new
                    FileOutputStream("c:\\temp.java",true));
              if (!t1.equals("void"))
              out.print("return "+t1.split(" ")[1]+";\n");
              out.print("}\n");
            }
   catch (Exception ex) { ex.printStackTrace();}
        }
 )
;
relation_name returns [String s]
```

```
{
        s=null;
}
: (RELATION ID {s=#ID.getText();})
;

input_type returns [String input]
{
        Type t;
        String s="";
        input="";
}
:(REFERS {System.out.print(" ");}
(t=object_type ID {s=#ID.getText();input=input+t.name +" "+s+","; }
)* {input=input.substring(0,input.length()-1);}
);

output_type returns [String p]
{
        p=null;
        Type t;
}
:(RETURNS  t=object_type ID   {p=t.name+" "+#ID.getText();})
;
facts
{
}
: (FORMULA  statements)
;

statements
{
}
:#(FACT_LIST  {

        try {
                    PrintStream out =new PrintStream(new
                    FileOutputStream("c:\\temp.java",true));
                   out.print("\n");
           }
   catch (Exception ex) { ex.printStackTrace();}

           } (statement)*)
```

```
;
attribute returns [String decl]
{
          Type t;
          String s;
          decl=null;
}


: ID   t=object_type {decl=t.name+" h_"+#ID.getText();} s=def_value[t] (STRING)?
         {
        try {
                              PrintStream out =new PrintStream(new
                              FileOutputStream("c:\\temp.java",true));
                               out.print("public "+t.name+" "+#ID.getText()+"="+s+";\n");
                               out.print("public  void"+"
                              set"+Util.initCap(#ID.getText())+"("+decl+"){\n"+#ID.getTex
                              t()+" =h_"+#ID.getText()+";\n}\n");
         }
        catch (Exception ex) { ex.printStackTrace();}

           }
            ;
derived_attribute
{
          Type t=null;
          String s="";
}
: DERIVED ID   t=object_type  s=def_value[t] (STRING)?
         {
          if (t.name.equalsIgnoreCase("double"))
          {
          try {
                                        PrintStream out =new PrintStream(new
                                        FileOutputStream("c:\\temp.java",true));

              out.print("public "+
                                        t.name+" "+#ID.getText()+"="+s+";\n");
                                       out.print("public  String"+"
                                       derive_"+Util.initCap(#ID.getText())+"(){\n");
              out.print("String txt,s1,s2,s3;\n");
                                                    out.print("NumberFormat formatter = new
                                        DecimalFormat(\"000000\");\n");
                                       out.print("formatter = new
                                       DecimalFormat(\"0.000\");\n");
              out.print("s1=");

                                        out.print("\""+Util.initCap(#ID.getText())+"="+s.repla
                                        ce("Math.","")+"\";\n");
                         out.print("s2=");

                                        out.print("\""+Util.initCap(#ID.getText())+" of
                                        "+"\"+");



                                        out.print("this.getClass().getName().toLowerCase()" +
                                        "+\" \""+   "+" + "name"+ "+\"=\"+");
                         out.print("formatter.format("+s+");\n");
                                        out.print("txt=s1+" + "\"\\n\""+"+"+"s2;\n");
                         out.print("return txt;\n"+"}\n");

                         }
                   catch (Exception ex) { ex.printStackTrace();}
          }
                         else if (t.name.equalsIgnoreCase("string"))
          {
          try {
              PrintStream out =new PrintStream(new
              FileOutputStream("c:\\temp.java",true));

                                        out.print("public "+t.name+"
                                        "+#ID.getText()+"="+s+";\n");
                                        out.print("public  String"+"
                                        derive_"+Util.initCap(#ID.getText())+"(){\n");
                         out.print("String txt,s1,s2,s3;\n");
              out.print("s1=");

                                        out.print("\""+Util.initCap(#ID.getText())+"=\"+"+s.r
                                        eplace("Math.","")+";\n");
                         out.print("s2=");
                                        out.print("\""+Util.initCap(#ID.getText())+" of
                                        "+"\"+");
                                        out.print("this.getClass().getName().toLowerCase()" +
                                        "+\" \""+   "+" + "name"+ "+\"=\"+");
                         out.print(s+";\n");
                                        out.print("txt=s1+" + "\"\\n\""+"+"+"s2;\n");
```

```
                out.print("return s1;\n"+"}\n");

                }
            catch (Exception ex) { ex.printStackTrace();}
             }
        else
                {
        try {
                                    PrintStream out =new PrintStream(new
                                    FileOutputStream("c:\\temp.java",true));

                                    out.print("public "+t.name+"
                                    "+#ID.getText()+"="+s+";\n");
                                   out.print("public  String"+"
                                   derive_"+Util.initCap(#ID.getText())+"(){\n");
                out.print("String txt,s1,s2,s3;\n");
                out.print("s1=");


                                   out.print("\""+Util.initCap(#ID.getText())+"="+s.repla
                                   ce("Math.","")+"\";\n");
                out.print("s2=");
                                  out.print("\""+Util.initCap(#ID.getText())+" of
                                  "+"\"+");

                                  out.print("this.getClass().getName().toLowerCase()" +
                                  "+\" \""+   "+" + "name"+ "+\"=\"+");
                out.print("\"todo\";\n");
                                   out.print("txt=s1+" + "\"\\n\""+"+"+"s2;\n");
            out.print("return txt;\n"+"}\n");

                }
        catch (Exception ex) { ex.printStackTrace();}
         }


        }

  ;
def_value[Type t] returns [String s]
{
        s=null;
}
```

```
: NUMBER { s=#NUMBER.getText();}
 | LBRAC it1:NUMBER {s="new "+t.name+"("+it1.getText();}   (COMMA
it2:NUMBER {s=s+","+it2.getText();})* {s=s+")";}
 |  t1:ID {s="new "+t.name+"("+t1.getText();}   (COMMA t2:ID
{s=s+","+t2.getText();})* {s=s+")";}
 | STRING {s="\""+#STRING.getText()+"\"";}
 | EXPR  STRING {s=#STRING.getText();}
 ;

statement
 {
         String s="";
 }
 : s=lvalue  ASSIGN rvalue {
                  try {
                  PrintStream out =new PrintStream(new
FileOutputStream("c:\\temp.java",true));

               out.print(";\n");
                  }
        catch (Exception ex) { ex.printStackTrace();}

         } ;

 lvalue returns [String s]
 {
         s="";
 }
 : (t1:ID {
               s=t1.getText();
                  try {
                                  PrintStream out =new PrintStream(new
                                  FileOutputStream("c:\\temp.java",true));

                         out.print(  t1.getText());
              }
                  catch (Exception ex) { ex.printStackTrace();}
 }                                (DOT t2:ID {
                         try {
                                  PrintStream out =new PrintStream(new
                                  FileOutputStream("c:\\temp.java",true));

                         out.print( "."+ t2.getText());
```

```
          }
         catch (Exception ex) { ex.printStackTrace();}
          })*      {
                    try {
                                PrintStream out =new PrintStream(new
                                FileOutputStream("c:\\temp.java",true));

                              out.print("=");
                    }
         catch (Exception ex) { ex.printStackTrace();}
                    })

;

rvalue
{
          String s,s1,s2;
}
:(s1=value {
                    try {
                              PrintStream out =new PrintStream(new
                              FileOutputStream("c:\\temp.java",true));

                             out.print(s1);
                  }
        catch (Exception ex) { ex.printStackTrace();}

                  }
//(OPERATOR {System.out.print(#OPERATOR.getText());} s1=value)?
(OPERATOR   s2=value {
                    try {
                                PrintStream out =new PrintStream(new
                                FileOutputStream("c:\\temp.java",true));

                              out.print(#OPERATOR.getText()+s2);
             }
        catch (Exception ex) { ex.printStackTrace();}

                             })*)
| EXPR STRING {
                    try {
                                PrintStream out =new PrintStream(new
                                FileOutputStream("c:\\temp.java",true));
                                out.print(#STRING.getText());
                  }
          catch (Exception ex) { ex.printStackTrace();}

                  }

        ;


var_list
{
}
:ID (ID)*;

object_type returns [Type t]
{
           t=null;
}
: NUM {t=new Type("double",100); }
|TEXT {t=new Type("String",100);}
|ID {t=new Type(Util.initCap(#ID.getText()),100);};

value returns [String s]
{
          s="UNKNOWN";
}
:CONSTANT {s=#CONSTANT.getText(); s=s.toUpperCase(); s="Math."+s;}
|ID {s=#ID.getText();}
|STRING {s=#STRING.getText();}
|NUMBER {s=#NUMBER.getText();};


/*************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Library g++
File: Id.java
*************************************************************/

public class Id {
          private String h_token;
          private Type h_t;
          public Id(String token,Type t){
                  h_token=token;
                  h_t=t;

          }
}
```

```
/***************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Library g++
File: Interpreter.java
***************************************************************/


import java.util.*;
import java.io.*;
import antlr.CommonAST;
import antlr.collections.AST;
import antlr.RecognitionException;
import antlr.TokenStreamException;
import antlr.TokenStreamIOException;


class Interpreter {
    SymbolTable symt;

  final static int fc_none = 0;
  final static int fc_break = 1;
  final static int fc_continue = 2;
  final static int fc_return = 3;

  private int control = fc_none;
  private String label;

  private Random random = new Random();

  public Interpreter() {
    symt = new SymbolTable( null, null );

  }
```

```
   public Type[] convertExprList( Vector v ) {
      /* Note: expr list can be empty */
        Type[] x = new Type[v.size()];
      for ( int i=0; i<x.length; i++ )
         x[i] = (Type) v.elementAt( i );
      return x;
   }

   public static String[] convertVarList( Vector v ) {
      /* Note: var list can be empty */
      String[] sv = new String[ v.size() ];
      for ( int i=0; i<sv.length; i++ )
         sv[i] = (String) v.elementAt( i );
      return sv;
   }

   public static Type getNumber( String s ) {
   Type t=new Type("double",100);
   return t;
   }


}


/*************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Library g++
File: SGMLDataType.java
*************************************************************/

import java.io.PrintWriter;
public class SGMLDataType
{
   String name;   // used in hash table

   public SGMLDataType() {
      name = null;
   }

   public SGMLDataType( String name ) {
      this.name = name;
```

```
}

public String typename() {
   return "unknown";
}

public SGMLDataType copy() {
   return new SGMLDataType();
}

public void setName( String name ) {
   this.name = name;
}

public SGMLDataType error( String msg ) {
   throw new SGMLException( "illegal operation: " + msg
                  + "( <" + typename() + "> "
                  + ( name != null ? name : "<?>" )
                  + " )" );

}

public SGMLDataType error( SGMLDataType b, String msg ) {
   if ( null == b )
      return error( msg );
   throw new SGMLException(
      "illegal operation: " + msg
      + "( <" + typename() + "> "
      + ( name != null ? name : "<?>" )
      + " and "
      + "<" + typename() + "> "
      + ( name != null ? name : "<?>" )
      + " )" );
}

public void print( PrintWriter w ) {
   if ( name != null )
      w.print( name + " = " );
   w.println( "<undefined>" );
}

public void print() {
   print( new PrintWriter( System.out, true ) );
```

```
    }

    public void what( PrintWriter w ) {
        w.print( "<" + typename() + ">  " );
        print( w );
    }

    public void what() {
        what( new PrintWriter( System.out, true ) );
    }

    public SGMLDataType assign( SGMLDataType b ) {
        return error( b, "=" );
    }


    public SGMLDataType uminus() {
        return error( "-" );
    }


    public SGMLDataType plus( SGMLDataType b ) {
        return error( b, "+" );
    }

    public SGMLDataType add( SGMLDataType b ) {
        return error( b, "+=" );
    }

    public SGMLDataType minus( SGMLDataType b ) {
        return error( b, "-" );
    }


}

/*************************************************************
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Library g++
File: SGMLSymbolTable.java
*************************************************************/

import java.util.*;
import java.io.PrintWriter;


//class SGMLSymbolTable extends HashMap {
public class SymbolTable {
private Hashtable table;
protected SymbolTable outer;
SymbolTable static_parent, dynamic_parent;
boolean read_only;

public SymbolTable( SymbolTable sparent, SymbolTable dparent ) {

static_parent = sparent;

dynamic_parent = dparent;
 read_only = false;
table=new Hashtable();
}


public SymbolTable(SymbolTable st) {
table=new Hashtable();
outer=st;
}
public void put (String token, Type t){
    table.put(token, new Id(token,t));
}
public Id get(String token){
    for (SymbolTable tab=this; tab!=null; tab=tab.outer){
    Id id=(Id)(tab.table.get(token));
if (id!=null) return id;
}
return null;
}
}
```

/***
Author: Sanjay Kumar sanjayk98@gmail.com
Date: 15-Dec-2006
Module: Library g++
File: Util.java
***/

```
class Util {

        public static String parse (String in)
        {
                String[] s;
                String p="";
                String o="";

                int i=0;
                s=in.split(",");
                while (i<s.length){
                        p=s[i];
                        p=p.split(" ")[1];
                        p=p.substring(2,p.length())+"="+p+";\n";
                        o=o+p;
                        i++;
                }
                return o;
        }

        public static String initCap (String in)
        {
                if (in == null || in.length() == 0)
                        return in;
                //
                boolean capitalize = true;
                char[] data = in.toCharArray();
                for (int i = 0; i < data.length; i++)
                {
                        if (data[i] == ' ' ||   Character.isWhitespace(data[i]))
                                capitalize = true;
                        else if (capitalize)
                        {
                                data[i] = Character.toUpperCase (data[i]);
                                capitalize = false;
                        }
                        else
                                data[i] = Character.toLowerCase (data[i]);
                }
                return new String (data);
        }       //      initCap
```