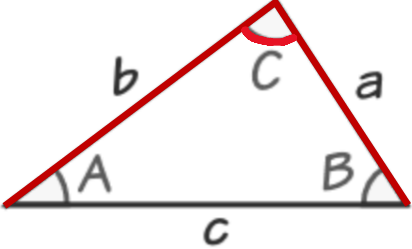


Lab 5

Create a new Eclipse project named **YourStudentId_OOP_Lab5** and add a class named **Triangle** to the project. You must comply with the following variable names.

1. Applications of Triangle (Write in Triangle class)

The class mainly asks students to build the *side*, *perimeter*, *area*, and *angle* method to process the side length and include angle of the two sides of a known triangle to calculate the other side length, perimeter, area, and the other two included angles.

Triangle Class	
<p>Attributes:</p> <p>sideA, sideB, sideC, angleA, angleB, angleC, perimeter, area</p> <p>Public Interface:</p> <p>side(3 parameters)</p> <p>perimeter(3 parameters)</p> <p>area(4 parameters)</p> <p>angle(3 parameters)</p>	

a. In the *main* method

- Declared the Triangle object named *triangle*
- Use Scanner object named *sideScanner* to read the sides of the triangle and assign to the *sideA*, and *sideB* attributes.
- Use Scanner object named *angleScanner* to read included angle of the sides and assign to the *angleC* attribute.
- Call the *side* method and assign the return value to the *sideC* attribute.
- Call the *perimeter* method and assign the return value to the *perimeter* attribute.
- Call the *area* method and assign the return value to the *arear* attribute.
- Call the *angle* method twice and assign the return values to the *angleA*, and *angleB* attribute.
- Print out the result as sample output.

b. In the *side* method

- Use *the law of cosines* to find the length of the other side based on the known length of the two sides and the enclosed angle.
- Declare a local variable named *sideC* and assign the calculation result to the variable.

c. In the *perimeter* method

- Sum the lengths of the three sides of the triangle
- Declare a local variable named *perimeter* and assign the calculation result to the variable.

d. In the *area* method

- Use the *Heron's formula* to calculate the area of the triangle based on the sides.
- Declare a local variable named *area* and assign the calculation result to the variable.

e. In the *angle* method

- Use *the law of cosines* to infer the diagonal angle.
- Declare a local variable named *angle* and assign the calculation result to the variable.

Sample output:

Please enter the length of the two sides of the triangle: 3 4
Please enter the angle of the triangle: 90
The other side length is 5
The perimeter of the triangle is 12
The area of the triangle is 6.00
The opposite angle of sideA(length(s):3.00) in the triangle is 37 degrees
The opposite angle of sideB(length(s):4.00) in the triangle is 53 degrees

API reference:

Modifier and Type	Method and Description
static int	abs(int a) Returns the absolute value of an int value.
static int	round(double a) Returns the closest int to the argument, with ties rounding up.
static double	pow(double a, double b) Returns the value of the first argument raised to the power of the second argument.
static double	sqrt(double a) Returns the correctly rounded positive square root of a double value.
static double	cos(double a) Returns the trigonometric cosine of an angle.
static double	toRadians(double angdeg) Converts an angle measured in degrees to an approximately equivalent angle measured in radians.
static double	toDegrees(double angrad) Converts an angle measured in radians to an approximately equivalent angle measured in degrees.
static double	acos(double a) Returns the arc cosine of a value; the returned angle is in the range 0.0 through pi.

Submission: Submit your project as “.zip file” via Moodle. No other submissions will be graded.

Reminder: Please zip **the whole project**

Deadline: Tomorrow’s midnight (for both Mon56 and Tue23)