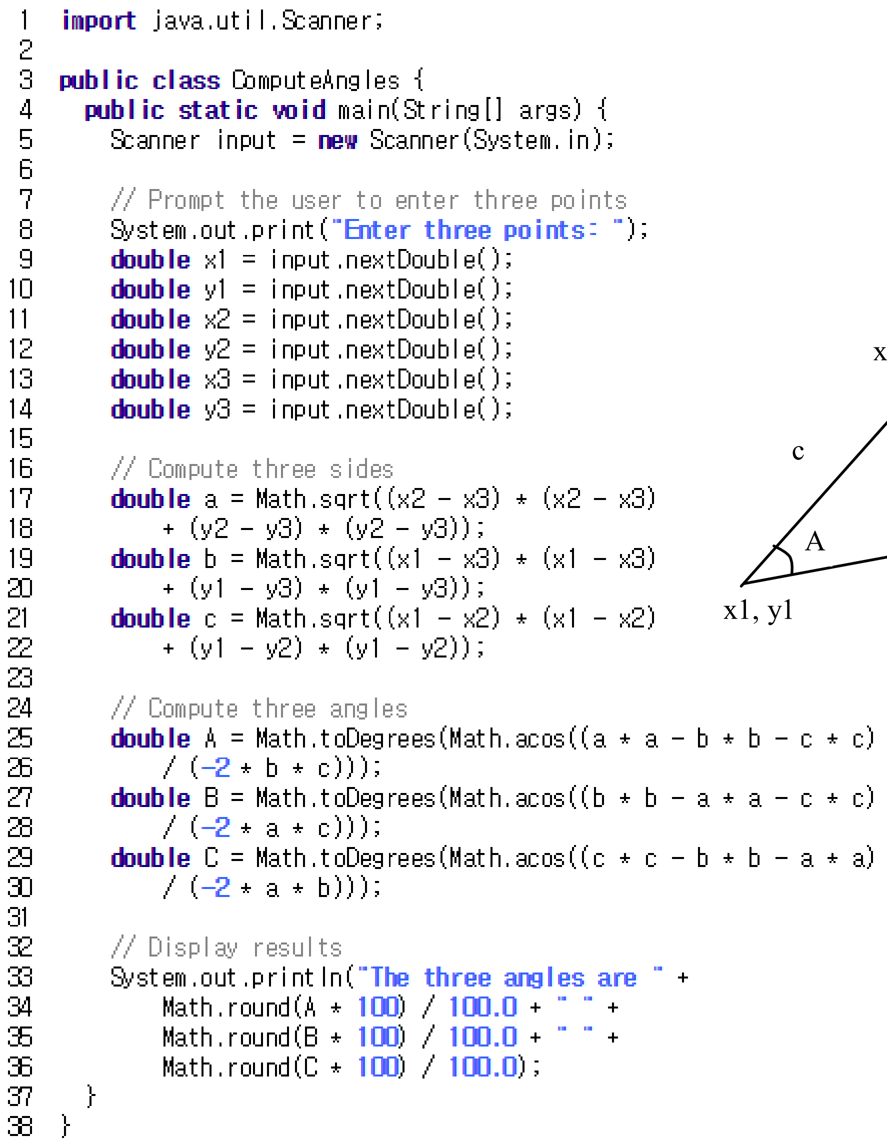
Exercise 4

Chapter 4 Mathematical Functions, Characters, and Strings

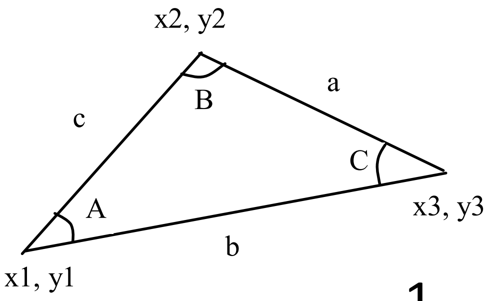
COMP217
Java Programming
Spring 2019

Instructor: Gil-Jin Jang

Ex 4-1 Angle of a Triangle



```
import java.util.Scanner;

public class ComputeAngles {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    // Prompt the user to enter three points
    System.out.print("Enter three points: ");
    double x1 = input.nextDouble();
    double y1 = input.nextDouble();
    double x2 = input.nextDouble();
    double y2 = input.nextDouble();
    double x3 = input.nextDouble();
    double y3 = input.nextDouble();

    // Compute three sides
    double a = Math.sqrt((x2 - x3) * (x2 - x3)
        + (y2 - y3) * (y2 - y3));
    double b = Math.sqrt((x1 - x3) * (x1 - x3)
        + (y1 - y3) * (y1 - y3));
    double c = Math.sqrt((x1 - x2) * (x1 - x2)
        + (y1 - y2) * (y1 - y2));

    // Compute three angles
    double A = Math.toDegrees(Math.acos((a * a - b * b - c * c)
        / (-2 * b * c)));
    double B = Math.toDegrees(Math.acos((b * b - a * a - c * c)
        / (-2 * a * c)));
    double C = Math.toDegrees(Math.acos((c * c - b * b - a * a)
        / (-2 * a * b)));

    // Display results
    System.out.println("The three angles are " +
        Math.round(A * 100) / 100.0 + " " +
        Math.round(B * 100) / 100.0 + " " +
        Math.round(C * 100) / 100.0);
  }
}
```



```
A = acos((a * a - b * b - c * c) / (-2 * b * c))
B = acos((b * b - a * a - c * c) / (-2 * a * c))
C = acos((c * c - b * b - a * a) / (-2 * a * b))
```

1. Write a Java program that prompts the user to enter the x- and y-coordinates of the three corner points in a triangle and then displays the triangle’s angles.
2. Translate the above in C

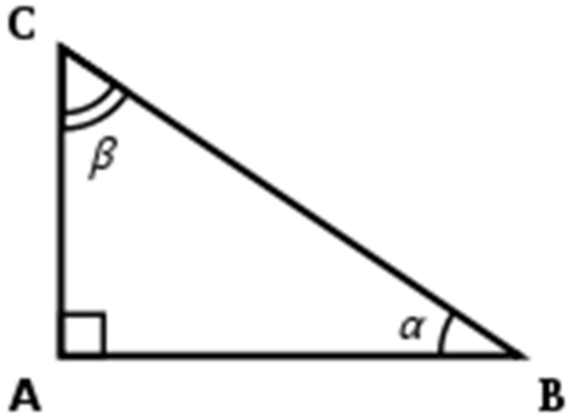
Submission:
ComputeAngles.java, ComputeAngles.c

*If you cannot find “toDegrees()” function in C, simply make one.

Ex 4-2 Computing Pythagorean Angles

- Write a Java program that computes the two angles of a right-angled triangle whose perpendicular edge lengths are 3 and 4
  - In the figure, AB = 4, AC = 3
  - Find the angles α and β in radians and degrees
- Translate it into C
- Submission:
  - ***PythagoreanAngles.java, PythagoreanAngles.c***

```
$ javac PythagoreanAngles.java
$ java PythagoreanAngles
alpha = 0.64 radians, 36.9 degrees
beta = 0.93 radians, 53.1 degrees
```



Ex 4-3 More Math Exercises

- Write java and C programs to compute the following equations:
  1. $\tan^{-1}(\sin 32° \cos 32°)$ (answer: 0.422352376952066, maybe shorter)
  2. $\sqrt[4]{e^{1.67} 1.65^{\log_{20} 8}}$ (answer: 1.655993893545619 , maybe shorter)
  3. $\log_7 21^{\ln 35}$
  4. $e^{\sqrt{\ln 4}+\sqrt{\ln 5}}$
- Submission (1 java, 1 C code)
  - Insert the above 4 equations in a single file
  - ***Ex43_math_exercises.java***
  - ***Ex43_math_exercises.c***

Ex 4-4 Random Number Generation

- Write Java and C program that generates random numbers between -18 and 9 (and including both), and displays the random number counts for the first appearances of those numbers

```
$ javac Ex44_RandomNumberGen.java
$ java Ex44_RandomNumberGen
>> Found -18 at 3
>> Found -18 at 23
>> Found -18 at 26
>> Found -18 at 29
>> Found 9 at 36
The first appearance of -18 was at 3
The first appearance of 9 was at 36
```

- Submission (1 java, 1 C code)
  - ***Ex44_RandomNumberGen.java*** and ***Ex44_RandomNumberGen.c***

- Java random number
  - 'double Math.random()' method returns a double random number in 0 and 1 (*not including 1*)
- C random number
  - See https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm
  - 'int rand()' returns an integer in 0 to RAND_MAX (very large, such as 2147483647)
    - Modulus (%) operation may be necessary
    - This function is a pseudo random generator from a fixed seed, so it may always generate the same number
  - Use 'void srand(unsigned int seed)' to set seed by the current time. For example

    ```
    #include <time.h>
    …
    srand((unsigned) time(NULL));
    ```

    - See the link in the above for details

Ex 4-5 Counting the number of spaces

- Write Java and C program that reads a line from the console, and count the number of spaces. Some part of Java code is given as follows:

```
String s = input.nextLine();
for (int i=0; i<s.length(); i++)
  if ( s.charAt(i) == ' ' ) num_spaces++;
/* execution example: (user input underlined)
Enter a line: Welcome to the world of Java
Number of spaces: 5
*/
```

- C coding hint: use ‘fgets(char *s, int size, FILE *stream)’
  - The similar function ‘gets(char *s)’ is deprecated (out of standard), so not recommended.
- Submission (1 java, 1 C code)
  - ***Ex45_CoutingSpaces.java*** and ***Ex45_CoutingSpaces.c***